차 례

머 리 말

위대한 령도자 **김정일**원수님께서는 다음과 같이 지적하시였다.

《프로그람을 개발하는데서 기본은 우리 식의 프로그람을 개발하는것입니다. 우리는 우리 식의 프로그람을 개발하는 방향으로 나가야 합니다.》

프로그람기술을 빨리 발전시키는것은 나라의 정보기술을 높은 수준에 올려세우기 위한 중요한 요구의 하나이다.

위대한 령도자 **김정일**원수님의 현명한 령도에 의하여 오늘 우리 나라에서는 정보기술을 빨리 발전시킬수 있는 물질기술적토대가 튼튼히 마련되였으며 새로운 정보기술성과들이 련이어 이룩되고있다.

나라의 정보기술을 높은 수준에 올려세우는데서 중요한것은 우리 식 조작체계 《붉은별》에서 응용할수 있는 프로그람들을 빨리 발전시키는것이다.

콤퓨터기술을 소유하는것은 사회성원으로서 응당 갖추어야 할 필수적인 자질로 된다. 여기서 프로그람지식이 차지하는 몫이 매우 크다.

프로그람작성방법을 알아야 콤퓨터가 어떻게 동작하는가 하는 원리를 더 잘 파악할수 있으며 인민경제의 정보화를 하루빨리 실현해나갈수 있다.

프로그람을 작성한다는것은 풀려는 문제를 잘 파악해서 그 풀이순서와 방법을 결정한 다음 그것을 콤퓨터가 아는 프로그람언어로 서술해주는것을 말한다.

5학년 《콤퓨터》에서는 Gambas에 의한 프로그람작성에서 가장 중요한 부분인 조종체들, 명령문들, 도형편집기능을 리용하여 여러가지 응용프로그람들을 작성하는 초보적인 방법에 대하여 배운다. 그리고 3차원화상제작도구인 3ds max를 리용하여 콤퓨터에서 3차원공간물체들을 모형화하고 동화상작품을 만드는 방법에 대하여 배우게 된다.

우리는 콤퓨터에 대한 학습을 열심히 하여 우리의 사회주의강성대국을 과학과 기술로 떠메고나갈 유능한 인재로 튼튼히 준비함으로써 위대한 령도자 **김정일**원수님의 크나큰 믿음과 기대에 보답하여야 한다.

제 1 장. Gambas프로그람작성

《붉은별》환경에서 우리는 구체적인 조작탁지령들을 몰라도 화면에 표시되여있는 창문, 그림기호, 차림표, 단추, 대화칸 등의 직관적인 도형요소들을 선택함으로써 콤퓨터와의 대화를 유연하게 할수 있다.

사용자와 콤퓨터와의 대화를 보장해주는 부분을 대면부(Interface)라고 하며 사용자와 콤퓨터사이의 대화를 그림기호를 리용하여 처리하여주는 직관적인 대면부를 도형사용자대면부(GUI: Graphical User Interface)라고 한다.

많은 경우 사용자들은 도형사용자대면부를 가지는 프로그람을 요구한다.

Gambas프로그람작성환경에서는 조종체(Control)들을 리용하여 도형사용자대면부를 가지는 시각적프로그람(Visual Program)을 손쉽게 작성할수 있다.

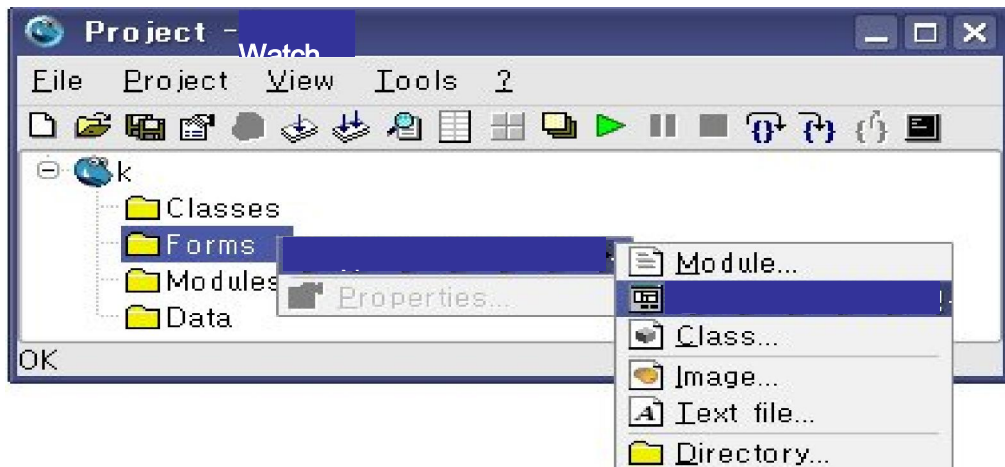
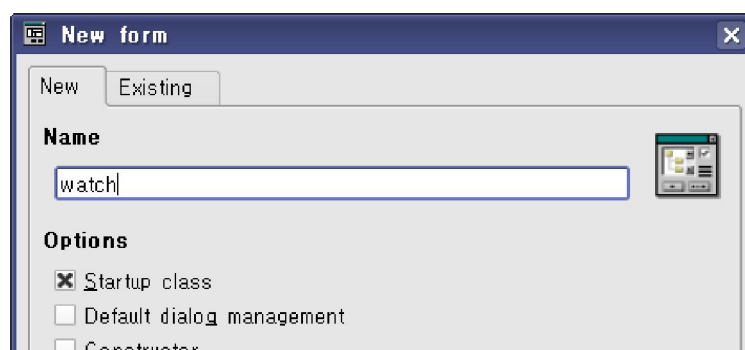
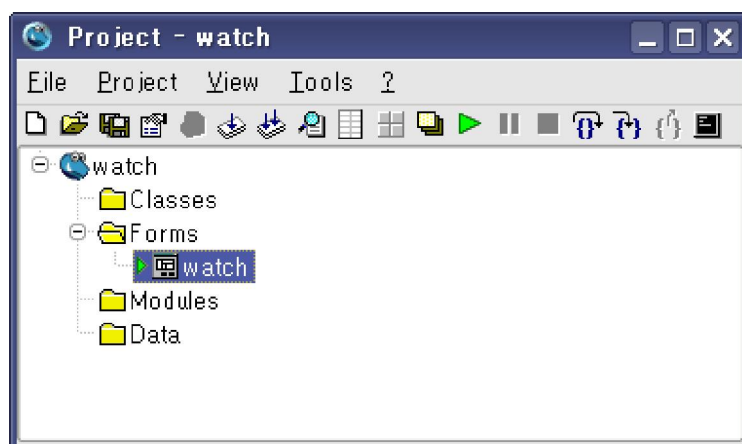
능률적인 프로그람을 작성하자면 프로그람코드를 재치있게 작성하는것과 함께 도형사용자대면부를 잘 설계하여야 한다.

제 1 절. 전자시계만들기

《전자시계》프로그람을 만드는 과정을 따라해보면서 조종체에 대한 개념과 도형사용자대면부프로그람작성방법을 익히도록 하자.

Gambas를 기동하고 새로운 프로젝트를 창조할 때 Create a graphical project항목을 선택한다.

프로젝트의 이름과 제목을 watch라고 입력한다.

기본창문에서 Forms를 선택하고 오른쪽찰칵하여 나타나는 지름차림표에서 New/ Form 지령을 실행하여 새로운 Form을 추가한다.

이때 나타나는 New form대화칸에서 Form의 이름을 watch라고 입력한다.

OK단추를 찰칵하면 watch Form이 추가된것을 볼수 있다.

대면부설계는 Form창문에서 한다.

도구칸(ToolBox)의 Form부분도구칸에서 단추조종체(Button)를 선택한다. 다음 Form창문에서 끌기로 4각형구역을 선택하면 단추조종체가 배치된다.

같은 방법으로 Form창문에 단추조종체 2개, 표식자조종체(Label) 1개, 본문칸조종체(TextBox) 1개를 배치한다.

다음은 속성칸(Properties)에서 조종체들의 속성을 설정한다.

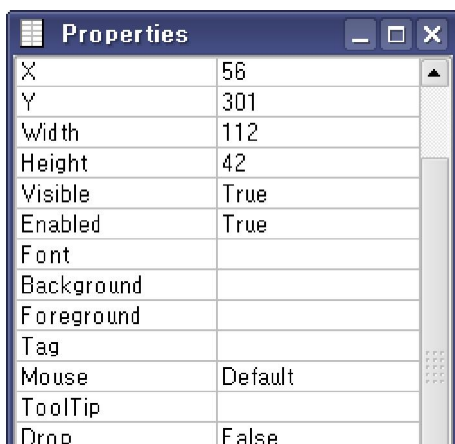
먼저 Form창문에서 첫번째 단추조종체를 선택한다. 그리고 속성칸에서 Text속성옆의 빈칸을 찰칵하여 유표가 나타나면 《날자보기》라는 문자렬을 써넣는다.

그러면 Form창문의 첫번째 단추조종체에 《날자보기》라는 글자가 나타난다.

같은 방법으로 두번째 단추조종체에 《시간보기》라는 글자를 쓴다.

다음 표식자조종체를 선택하고 속성칸에서 Text

Properties

X	56
Y	301
Width	112
Height	42
Visible	True
Enabled	True
Font	
Background	
Foreground	
Tag	
Mouse	Default
ToolTip	
Drop	False

속성값을 《전자시계》라고 입력한다.

Font속성칸의 작은 단추를 찰칵하여 나타나는 대화칸에서 적당한 서체와 크기를 지정한다.

마지막으로 본문칸조종체를 선택한다. 그리고 Text속성칸의 문자렬을 선택하고 건반의 Delete건을 눌러 지워버린다.

그러면 그림과 같은 대면부가 완성된다.

코드창문에서는 프로그람코드를 작성한다.

Form창문에서 단추조종체 《날자보기》를 두번찰칵한다. 그러면 코드창문으로 넘어

가면서 다음과 같은 사건수속이 나타난다.

```
PUBLIC SUB Button1_Click()
…
END
```

이것은 단추를 찰칵할 때 수행하는 코드를 쓰는 부분으로서 Gambas체계가 자체로 발생시킨다. 여기에 코드를 쓰고 실행시키면 단추를 찰칵할 때 그안에 있는 코드내용이 수행된다.

사건수속안에 다음과 같은 코드를 쓴다.

```
PUBLIC SUB Button1_Click()

TextBox1.Text=Date(Now)

END
```

단추조종체 《시간보기》를 두번찰칵하고 같은 방법으로 다음의 코드를 써넣는다.

```
PUBLIC SUB Button2_Click()

TextBox1.Text=Time(Now)

END
```

Project창문에서 자기가 실행하려고 하는 Form을 선택하고 오른쪽찰칵하여 나타나는 지름차림표에서 startup class지령을 실행한다. 그다음 F5건을 누르거나 도구띠의 Run단추를 찰칵하면 프로그람이 실행된다.

실행화면은 다음과 같다.

우리가 우에서 쓴 조종체들에 대하여 구체적으로 보기로 하자.

1. 조종체

1) 조종체의 종류

Gambas를 기동하면 항상 도구칸에 표준적으로 나타나는 조종체들이 있는데 이 조종체들을 표준조종체라고 한다. 도구칸은 3개의 부분도구칸 Form과 Container, Special부분으로 구성되여있다.

그림은 Form부분도구칸에 있는 조종체들을 보여주고 있다. 표준조종체들외에 다른 외부조종체들도 사용할수 있다.

외부조종체들을 사용하자면 Project/Properties를 찰칵하고 components를 선택한 다음 해당한 조종체묶음을 지적하면 된다.

2) 조종체의 속성

조종체의 속성을 보기 위해서는 다음의 조작을 수행한다.

우선 Form창문에 조종체를 배치하고 그것을 선택한 다음 속성칸에서 속성값을 설정한다.

속성칸에서 왼쪽렬은 속성의 이름이고 오른쪽렬은 속성값이다.

속성칸이 나타나지 않았을 때에는 기본창문의 View차림표에서 Properties지령을 실행시키든가 기능건 F4를 누르면 된다.

우리가 앞에서 리용한 단추조종체, 표식자조종체, 본문칸조종체들은 다같이 Text속성을 가지고있는데 이 속성은 실지 대면부에서 나타나는 조종체들의 제목이다.

속성설정은 코드작성단계에서도 할수 있으며 속성값은 값주기명령문을 리용하여 설정할수 있다.

형식:

조종체이름.속성이름=속성값

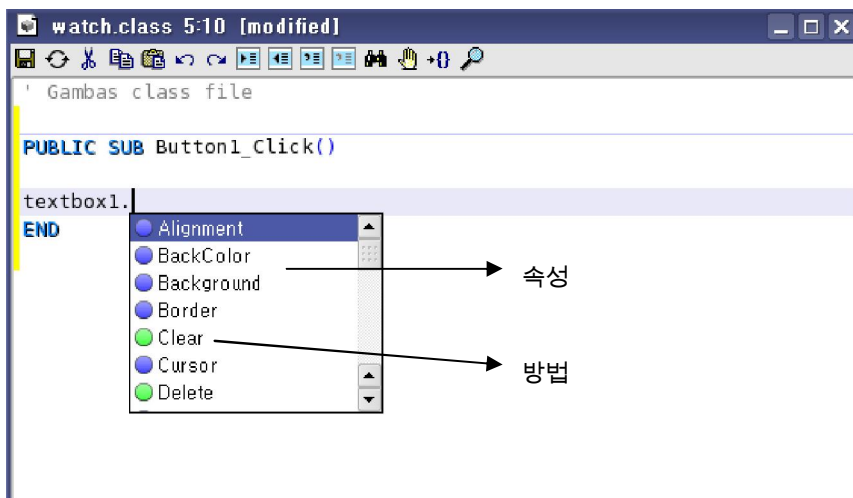
우의 실례에서 Text속성을 속성칸에서 설정하지 않은 경우에는 다음의 코드로 실현할수 있다.

```
PUBLIC SUB Form_Open()
Button1.Text="날자보기"
Button2.Text="시간보기"
```

Label1.Text="전자달력"

TextBox1.Text=" "

END

우의 코드를 코드칸에서 입력할 때 조종체이름을 쓰고 점을 찍으면 그림과 같이 내리펼침목록이 나타나는데 여기서 푸른색동그라미로 강조된 항목은 속성의 이름이다.

Name속성은 모든 조종체가 공통으로 가지고있는 속성으로서 코드작성에서 리용되는 조종체의 이름을 반영한다.

Name은 설계상태에서만 설정할수 있는 속성이다.

객체들의 이름은 조종체이름의 략자 3개를 앞에 붙이고 여기에 객체의 뜻을 의미하는 기호렬을 붙여짓는것이 좋다. 그래야 많은 코드안에서 객체변수들을 쉽게 구별할수 있다.

표식자조종체의 앞붙이략자는 lbl이다.

3) 조종체의 방법

조종체의 방법은 조종체에 대하여 그것을 보여주거나 숨기거나 움직이는것 등과 같이 그 무엇인가를 하도록 하는 고정된 프로그람이다.

조종체의 방법을 리용하자면 값주기명령문을 쓴다.

형식:

조종체이름.방법(인수1, 인수2,…)

우의 그림에서 풀색동그라미로 강조된 항목은 조종체의 방법이다.

실례로 본문칸조종체에는 Delete라는 방법이 있는데 이 방법은 조종체를 없앤다.

TextBox1.Delete

4) 조종체의 사건

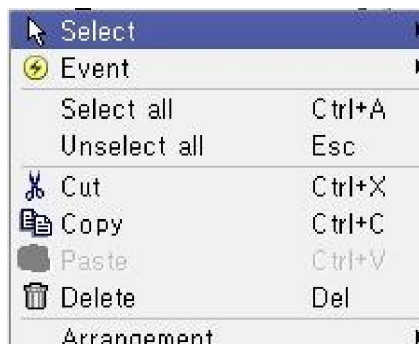
사용자나 조작체계가 조종체에 대하여 진행하는 작용을 사건이라고 한다.

실례로 《누른다》, 《초점을 준다》, 《열기한다》등은 사건이다.

사건은 프로그람동작의 계기로 된다.

매개 사건에 대하여 그것을 처리하기 위한 일련의 코드를 작성하는데 이것을 사건수속(Event procedure)이라고 한다.

코드칸에서 사건수속의 틀거리는 자동적으로 만들어진다.

조종체의 사건수속을 처리하려면 Form창문에서 조종체를 선택하고 오른쪽찰칵하였을 때 나타나는 지름차림표에서 Event를 선택하고 해당한 사건을 선택하면 된다.

조종체에 대한 개념

조종체는 객체를 만들기 위한 도구이다.

우리가 살고있는 현실세계의 모든 대상들을 객체라고 할수 있다.

모든 대상들은 고유한 속성을 가지고있으며 외적인 작용이 가해지면 어떤 행동으로 표현할 수 있는 능력을 가지고있다.

실례로 《공》이라는대상은 재질, 크기, 색갈, 튐성결수 등의 속성을 가지고있으며 그 속성값에서 다른 공들과 구별된다. 또한 《공》은 사람이 발로 차는 외적인 작용이나 손으로 던지는 외

2. 단추조종체

단추조종체 는 사용자가 어떤 처리를 하는 지령을 시작하거나 끝낼 때 리용하는 조종체이다.

- 속성

Name: 프로그람에서 리용되는 단추조종체의 이름을 반영한다.

설계상태에만 설정할수 있는 속성이다. 실례로 이 속성값을 btnstart로 하였다면 이 조종체를 두번찰칵하여 나타나는 코드창문에는 다음과 같은 사건수속이 표시된다.

```
PUBLIC SUB btnstart_Click( )
```

…

END

Text: 단추우에 나타나는 제목을 설정한다. Name속성과는 달리 단추의 제목만을 설정한다.

앞에서 설명한바와 같이 설계시와 코드작성시에 설정이 가능하다.

BackColor: 조종체의 배경색을 지정한다.

색값을 설정하기 위해서 미리 정해진 색갈상수값을 리용할수 있다.

례: Button1.Backcolor=Color.Red

단추의 색갈을 붉은색으로 설정한다.

Border: 조종체의 경계를 설정한다.

Cancel: 사용자가 건반에서 Esc건을 눌렀을 때 유연하게 탈퇴할수 있게 해준다.

• 방법

Delete: 조종체를 없앤다.

Hide/Show: 조종체를 숨기거나 보이게 한다.

3. 표식자조종체

표식자조종체(Label) **A** 는 화면에 본문을 표시하는 조종체로서 주로 다른 조종체의 설명문으로 리용된다.

• 속성

Name: 프로그람에서 리용되는 단추조종체의 이름을 반영한다. 설계상태에서만 설정할수 있는 속성이다.

Text: 본문의 표시는 설계시에 이 속성을 설정하여 할수 있으며 프로그람의 수행과정에는 사용자가 직접 입력할수 없다.

수행과정에 본문을 변경시키려면 코드를 통하여 속성값을 설정하여야 한다.

• 사건

DblClick: 조종체를 두번찰칵할 때 발생한다.

4. 본문칸조종체

본문칸조종체(TextBox) abc 는 한 행으로 된 본문자료를 입력하거나 표시하는데 리용되는 조종체이다.

본문칸조종체는 표식자조종체와는 달리 본문을 설계시에 입력하여 표시할수도 있고 실행시에 입력하여 표시된 본문을 변경할수도 있다.

• 속성

Text: 입력된 본문을 표시하거나 돌려주는 속성이다. 다시말하여 사용자가 본문칸에 입력한 본문내용을 가지고있는 속성 또는 이 속성에 값을 주면 실행시에 나타나는 속성이다.

속성도 하나의 변수이다. 따라서 자료형을 가진다. 이 속성의 자료형은 기호렬형이다. 그러므로 수값을 입력하려는 경우 본문칸으로 입력하였을 때에는 입력된 속성값을 다시 수값으로 변환하여 리용하여야 한다.

례: a=Val(txta.Text)

b=Val(txtb.Text)

Font: 조종체가 표시하는 본문의 서체이름, 서식, 크기를 규정하는 속성이다.

속성단추(...)를 찰칵하면 Select a font대화칸이 펼쳐진다.

Family목록에서 서체이름을, Style목록에서 서식을, Size목록에서 서체크기를 선택한다.

• 사건

KeyPress: 조종체가 초점을 가지는 동안 건이 눌리울 때 발생하는 사건이다.

본문칸에 기호 5개가 입력되면 이 사건은 5번 발생된다.

초점을 가진다는것은 Form창문안에서 조종체가 마우스나 건에 의하여 선택되여 실행되고있다는것을 말한다.

실 습

1. 《전자시계》 프로그람을 작성하여라.

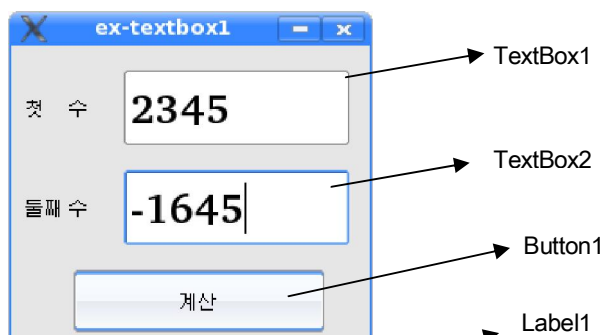
<대면부>

<코드>

```
PUBLIC SUB Button1_Click()

TextBox1.Text=Date(Now)

END

PUBLIC  SUB  Button2_Click()

TextBox1.Text=Time(Now)

END
```

2. 두 수의 합을 계산하는 프로그램을 작성하여라.

<대면부>

<속성>

조종체이름	속성이름	속성값
TextBox1	Name	txta
	Text	
TextBox2	Name	txtb
	Text	
Button1	Name	btncalc
	Text	계산
Label1	Name	lblresult

<코드>

```
PUBLIC SUB btncalc_Click()
```

```
DIM a AS Float, b AS Float, c AS Float
a=Val(txta.Text)          ①
b=Val(txtb.Text)          ②
c=a+b                     ③
lblresult.Text=Str(c)     ④
END
```

<코드설명>

btncalc_Click사건수속에서 a, b는 입력한 두 수를 받는 변수이다.

변수 c는 두 수의 합을 보관하는 변수이다.

①행: txta본문칸에 입력된 자료를 수값형으로 변환하여 변수에 넣는 코드이다.

②행: txtb본문칸에 입력된 자료를 수값형으로 변환하여 변수에 넣는 코드이다.

③행: 두 수를 더하여 결과변수 c에 넣는 과정이다.

④행: 변수 c의 내용을 문자렬로 변환하여 결과를 표식자에 출력하는것이다.

대면부에서 매개 본문칸이 어떤것인가를 표식자로서 설명하였다.

본문칸은 자료를 입력하여 변수에 넣는데 리용한다.

3. 《조선을 위하여 배우자!》구호를 두번찰칵하면 《Let's learn for Korea!》를 표시하는 프로그람을 작성하여라.

<대면부>

<속성>

조종체이름	속성이름	속성값
Label1	Name	lblkorea
	Foreground	붉은색
	Text	조선을 위하여 배우자!
Label2	Name	lbleng
	Text	

<코드>

```
PUBLIC SUB lblkorea_DblClick()

lbleng.Text="Let's learn for Korea!"        ①

END
```

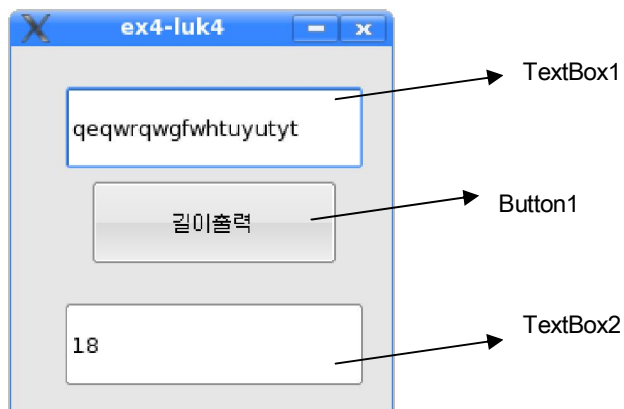
<코드설명>

코드의 lblkorea_DblClick사건은 《조선을 위하여 배우자!》구호를 표시한 표식자를 두번 찰칵할 때 수행된다.

코드의 ①행은 lbleng의 Text속성에 《Let's learn for Korea!》라는 기호렬을 설정하는 내용이다. 그러면 이 기호렬이 lbleng에 표시된다.

4. 본문칸에 기호렬을 입력하면 그의 길이를 출력하는 프로그람을 작성하여라.

<대면부>

<속성>

조종체이름	속성이름	속성값
TextBox1	Name	txtsmall

	Text	
TextBox2	Name	txtlen
	Text	
Button1	Name	btnout
	Text	길이출력

<코드>

```
PUBLIC SUB btnout_Click()
DIM s AS Byte                    ①
s=Len(txtsmall.Text)           ②
txtlen.Text=s                    ③
END
```

<코드설명>

코드에서 s변수는 txtsmall본문칸에 입력된 본문의 길이를 받는 변수이다.
②행의 Len함수는 문자렬의 길이를 구하는 함수이다.
코드의 ③행은 구한 문자렬의 길이를 txtlen본문칸에 출력하기 위한것이다.

련습문제

1. 두 수의 차, 적, 상을 계산하는 프로그람을 작성하여라.
2. 3각형의 밑변과 높이를 입력하면 면적을 계산하는 프로그람을 작성하여라.
3. 기호렬 두개를 입력하면 두개의 기호렬을 련결하여 출력하는 프로그람을 작성하여라.
4. 오늘의 날자를 본문칸에 출력하고 지령단추를 찰칵하면 주체년호로 변환하여 출력하는 프로그람을 작성하여라.
5. 본문칸에 기호들을 입력하면 다음 본문칸에 그의 아스키코드가 출력되게 하여라.
6. 공의 속도와 시간을 입력하면 등속운동하여 간 거리를 출력하는 프로그람을 작성하여라.

제 2 절. 《조선을 위하여 배우자!》

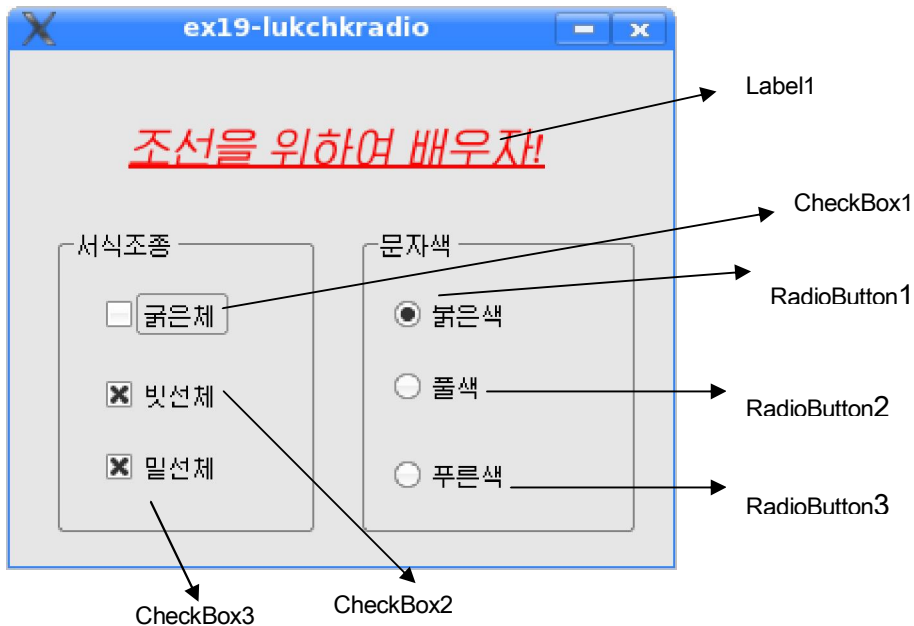
《붉은별》 조작체계에서 《화상편집기》의 《설정/도구띠들》 부분차림표에는 기본도구띠, 본문도구띠, 색칸, 도구칸과 같은 항목들을 표시하겠는가, 하지 않겠는가를 표시하는 검사칸들이 있다.

이 검사칸을 찰칵하여 검사기호가 표시되면 그 도구띠가 표시되거나 없어지는것을 볼수 있다.

《조선을 위하여 배우자!》라는 문자렬의 서식과 문자색을 조종하는 프로그람을 작성하면서 검사기호를 리용하여 항목에 대한 설정, 비설정을 하는 방법과 조건명령문에 의한 프로그람흐름조종방법을 익히도록 하자.

대면부설계를 위하여 도구칸의 Form부분도구칸에 있는 검사칸조종체(CheckBox)와 선택단추조종체(RadioButton)를 리용한다.

<대면부>

<속성>

조종체이름	속성이름	속성값
Label1	Name	lbl
	Text	조선을 위하여 배우자!
RadioButton1	Name	rdbred
	Text	붉은색
RadioButton2	Name	rdbgre
	Text	풀색
RadioButton3	Name	rdbblue
	Text	푸른색
CheckBox1	Name	chkbold
	Text	굵은체
CheckBox2	Name	chkitalic
	Text	빗선체
CheckBox3	Name	chkunder
	Text	밑선체

<코드>

```
PUBLIC SUB chkbold_Click()

IF chkbold.Value=True THEN

lbl.Font.bold=True

ELSE

lbl.Font.bold=False

ENDIF

END

PUBLIC SUB chkitalic_Click()

IF chkitalic.Value=True THEN

lbl.Font.italic=True

ELSE

lbl.Font.italic=False

ENDIF

END

PUBLIC SUB chkunder_Click()
```

```
IF chkunder.Value=True THEN

lbl.Font.underline=True

ELSE

lbl.Font.underline=False

ENDIF

END



PUBLIC SUB rdbred_Click()

lbl.Forecolor=Color.Red

END



PUBLIC SUB rdbgre_Click()

lbl.Forecolor=Color.Green

END



PUBLIC SUB rdbblue_Click()

lbl.Forecolor=Color.Blue

END
```

<코드설명>

코드에서 chkbold_Click, chkitalic_Click, chkunder_Click는 각각 굵은체, 빗선체, 밑선체를 표시하는 검사칸을 찰칵하였을 때 발생하는 사건들이다. 이 사건들에서는 검사칸들의 상태를 나타내는 Value속성값을 판정하여 True이면 lbl표식자의 글발의 서식을 변화시키게 되여있다.

검사칸들의 상태판정은 조건명령문 IF로 한다. 또한 rdbred_Click, rdbgre_Click, rdbblue_Click사건들은 각각 붉은색, 풀색, 푸른색을 나타내는 선택단추를 찰칵하였을 때 발생하는 사건이다. 이 사건들에서는 문자렬의 색을 나타내는 Forecolor속성을 Color객체의 색코드값으로 설정하여 색을 조종하게 되여있다.

코드를 실행시켜보면 우리가 《화상편집기》에서 본것처럼 검사칸을 선택하는데 따라 해당한 항목들이 설정된다는것을 알수 있다.

1. 조건명령문

설정한 문제에서는 굵은체를 선택하였는가, 하지 않았는가 또는 빗선체를 선택하였는가, 선택하지 않았는가 등의 조건에 따라 처리를 진행하도록 되여있는데 여기에 조건명령문을 리용하였다.

어떤 값의 상태를 판정하여 결과에 따라 동작을 선택하여 수행하게 하는 명령문이 바로 조건명령문이다.

우리는 일상생활에서 《오늘 날씨가 좋으면 수영하러 가자.》, 《공부를 잘하면 최우등을 할수 있다.》와 같이 어떤 전제조건에 따라서 동작을 선택하는 경우가 많다.

콤퓨터에서도 역시 변수나 식의 값을 판정하여 결과에 따라 서로 다른 명령문들을 실행하여야 할 경우가 매우 많다. 바로 이런 경우에 조건명령문을 리용한다.

<형식>

① IF 조건식 THEN 명령문

② IF 조건식1 THEN
명령문렬1
[ELSE IF 조건식2 THEN
명령문렬2]

[ELSE
명령문렬]
ENDIF

<기능>

① 조건식의 값이 True이면 명령문을 수행한다. 명령문은 한개만 쓸수 있다.

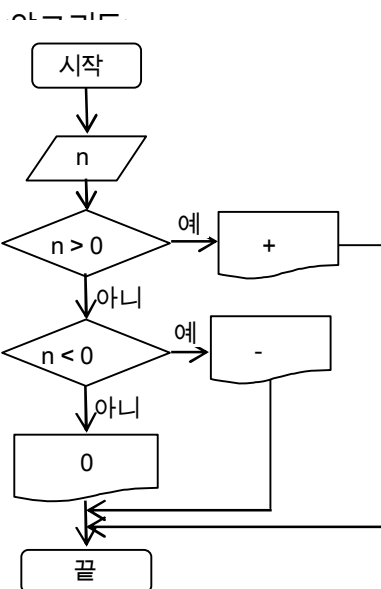
명령문수행이 끝나면 IF다음의 명령문을 수행한다.

② 조건식1의 값이 True이면 명령문렬1을 수행하고 False이면 ELSE IF다음의 조건식2를 판정한다. 역시 이 조건식의 값이 True이면 명령문렬2를 수행하고 False이면 다음조건을 판정한다.

그런 식으로 조건이 만족되는 경우의 명령문들을 집행한다. 만족되는 조건이 없으면 ELSE다음의 명령문들을 수행한다.

ELSE부분이 생략된 경우에 조건식중에 어느것도 만족되지 않으면 ENDIF다음의 명령문을 집행한다.

례: 임의의 수를 입력하여 령보다 크면 +, 작으면 -, 령이면 0을 출력하는 프로그람을 작성하여라.

〈코드〉

```
PUBLIC SUB btncalc_Click()

INPUT  n

IF n>0 THEN

PRINT "+"

ELSE IF n<0 THEN

PRINT "-"

ELSE

PRINT "0"

ENDIF
```

END

프로그람에서는 먼저 수값이 정수인가를 판단한다. 정수인 경우에는 +라고 출력하고 ENDIF문 다음행으로 이행한다.

정수가 아닌 경우에는 부수인가를 판단하고 부수이면 -를 출력하고 ENDIF문 다음행으로 이행하며 부수가 아니면 0을 출력하고 다음행으로 이행한다.

SELECT/CASE 명령문

IF 명령문과 함께 많이 쓰이는 조건명령문이다.

많은 값들중에서 하나의 값을 선택하여 여러가지 처리중 하나의 처리를 진행해야 하는 경우에 이 명령문을 리용한다.

명령문의 형식:

SELECT 비교식

[CASE 값 1 [값 2]]

2. 검사칸조종체

우리가 일상적으로 보면 최우등인가 아닌가, 날씨가 좋은가 나쁜가 등 두가지중 하나를 확인하려고 하는 경우가 많다. 이렇게 두가지 상태를 표시하는데 검사칸을 리용한다.

검사칸조종체(CheckBox) 는 선택/비선택의 두가지 값으로 다른 대상의 상태를 설정하게 하며 비교적 크지 않은 묶음에서 사용자가 몇개의 항목들을 동시에 선택하는 동작을 실현하기 위한 조종체이다.즉 검사칸조종체는 꼭같은 검사칸 여러개로 묶음을 만들면 그중에서 하나 또는 둘, 그이상의 항목들을 선택하여 목적하는 대상자료의 상태를 설정할수 있는 특성을 가지고있다. 검사칸 여러개를 묶는데는 틀조종체(Frame)가 리용된다.

- 속성

Value: 검사칸의 상태를 반영하는 속성으로서 론리형이다. 즉 속성값은 True나 False이다. 검사칸을 선택할 때마다 Value속성이 서로 바뀌는데 찰칵하여 검사기호가 생기면 이 속성값은 True로 설정되고 다시 찰칵하여 검사기호가 없어지면 False로 된다.

< > :True, < > :False

설정한 문제에서는 검사칸으로 본문의 서식을 설정하게 되여있다.

우리는 이미 본문칸조종체를 학습하면서 본문의 서체를 설계시 변화시키는 방법에 대하여 학습하였다. 문제에서는 실행시 검사칸의 동작으로 본문의 서식을 변화시켜야 하므로 코드에서 서체속성들을 설정하여야 한다.

실행시 본문의 서체변경을 해보자.

서체를 나타내는 속성은 Font이다. 이 속성은 서체이름, 서식, 크기 등을 속성으로 가지는 객체이다. 따라서 서체이름은 Font.name, 서식은 Font.bold, Font.underline, Font.italic속성으로 설정하며 크기는 Font.size속성으로 규정한다.

실행시 본문의 서체를 변화시키려면 Font의 이러한 속성값들을 코드로 설정하여야 하는데 서체이름은 기호렬형, 서체크기는 옹근수형, 서식을 나타내는 속성값들은 모두 론리형값이다.

```
객체.Font.name=서체이름
객체.Font.size=산수식
객체.Font.bold=True(혹은 False)
객체.Font.italic=True(혹은 False)
객체.Font.underline=True(혹은 False)
```

례: Label1.Font.bold=True
Label1.Font.italic=True

Label1.Font.underline=False

• 사건

지령단추조종체와 마찬가지로 Click사건이 많이 리용된다. Click사건으로 검사칸은 자동적으로 그것의 상태를 변화시킨다.

검사칸의 앞붙이략자는 chk로 한다.

3. 선택단추조종체

선택단추조종체(RadioButton) 역시 검사칸과 마찬가지로 선택/비선택으로 다른 대상의 상태를 두가지중 하나로 설정하는데 리용한다. 그런데 검사칸과 다른 점은 선택단추 여러개를 묶어 묶음으로 만들면 사용자가 이가운데서 한개의 항목만을 선택할수 있다는것이다. 즉 선택단추 하나를 선택하였다가 다른 선택단추를 선택하면 이미 선택되였던것이 자동적으로 해제된다. 따라서 선택단추묶음에서는 언제나 선택된것이 하나밖에 없다.

• 속성

Value: 선택단추조종체의 상태를 나타내는 속성으로서 론리형이다.

선택단추가 선택되였으면 True, 선택되지 않았으면 False이다.

Form창문에 선택단추 두개를 배치하고 그의 상태를 보자.

첫째 선택단추를 선택하였다가 다른 선택단추를 선택하면 첫째 선택단추가 해제되는것을 볼수 있다. 이것을 통하여 선택단추가 서로 배타적으로 작용한다는것을 알수 있다. 그러므로 묶음으로 된 선택단추들은 하나가 Value값이 True이면 다른것들은 모두 False로 된다.

Visible: 실행시에 객체의 현시상태를 규정한다. True이면 객체를 표시하고 False이면 표시하지 않는다.

Enabled: 조종체의 동작을 허가 또는 금지한다. 속성값은 론리형이다.

례: PUBLIC SUB Button_Click()

```
Chk1.Visible=False
Opt1.Enabled=False
END
```

이 실례는 단추를 찰칵할 때 검사칸은 보이지 않게 하고 선택단추는 동작을 금지시키도록 하는 코드이다.

우리가 설정한 문제에서는 선택단추를 리용하여 본문의 색을 조종하게 되여있다.

이것을 진행하려면 실행시 본문의 색을 규정하는 방법에 대하여 알아야 한다.

우리는 이미 표식자조종체에서 설계시 문자의 색을 규정하는 방법에 대하여 학습하였다. 그러면 실행시 조종체를 선택할 때 문자의 색이 변하게 하려면 어떻게 해야 하겠는가?

실행시에 다음과 같이 색을 규정하는 코드를 작성해넣어야 한다.

객체.Foreground=Color.색이름

여기서 Color는 색을 주는 객체이다.

색은 모두 Color객체의 속성으로 되여있다.

색이름은 영어로 색을 나타내는 단어를 쓰면 된다.

례: 검사칸의 문자색을 붉은색으로 변화시키려면 다음과 같이 쓸수 있다.

```
PUBLIC SUB Form_Open()
CheckBox1.Foreground=Color.Red
END
```

이 코드는 Form이 펼쳐질 때 검사칸의 문자색을 붉은색으로 실행시에 설정하게 하는것이다.

색상수(Color Constants)

체계상수	설 명
Black	검은색
Blue	푸른색
Green	풀색
Cyan	하늘색

• 사건

선택단추조종체에서도 검사칸조종체에서와 같이 Click사건이 많이 리용된다.

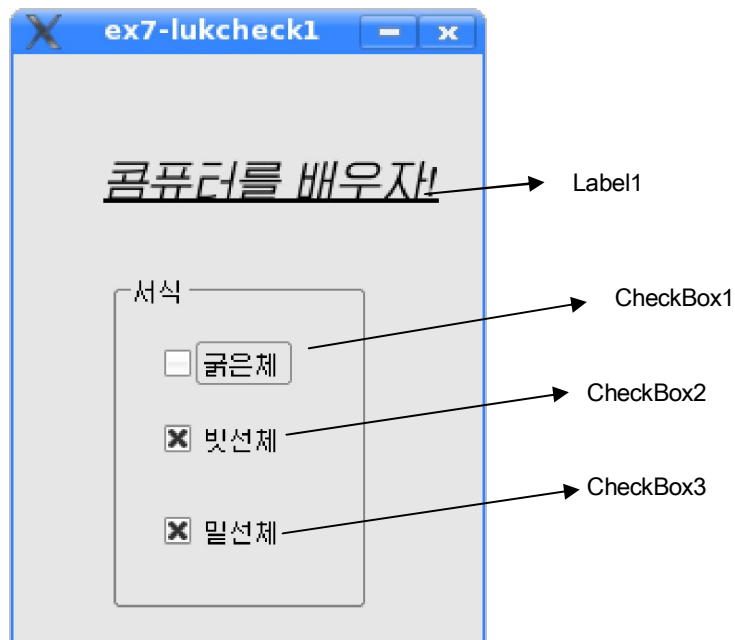
Click사건으로 선택단추는 자동적으로 그것의 상태를 변화시킨다.

선택단추조종체의 앞붙이략자는 rdb로 한다.

실 습

1. 《콤퓨터를 배우자!》라는 문자렬의 서식을 조종하는 프로그람을 작성하여라.

<대면부>

<속성>

조종체이름	속성이름	속성값
Label1	Name	lblcom
	Text	콤퓨터를 배우자!
CheckBox1	Name	chkbold
	Text	굵은체
CheckBox2	Name	chkitalic
	Text	빗선체
CheckBox3	Name	chkunder

	Text	밑선체

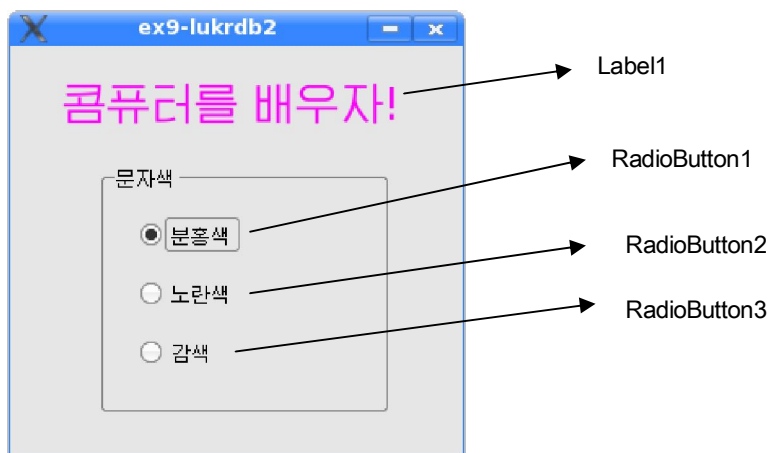
<코드>

```
PUBLIC SUB chkbold_Click()
IF chkbold.Value=True THEN
lblcom.Font.bold=True
ELSE
lblcom.Font.bold=False
ENDIF
END

PUBLIC SUB chkitalic_Click()
IF chkitalic.Value=True THEN
lblcom.Font.italic=True
ELSE
lblcom.Font.italic=False
ENDIF
END
```

```
PUBLIC SUB chkunder_Click()

IF chkunder.Value=True THEN

lblcom.Font.underline=True

ELSE

lblcom.Font.underline=False

ENDIF

END
```

2. 문자색을 나타내는 칸안에서 선택하는 색에 따라 《콤퓨터를 배우자!》라는 문자렬의 색이 조종되게 하여라.

<대면부>

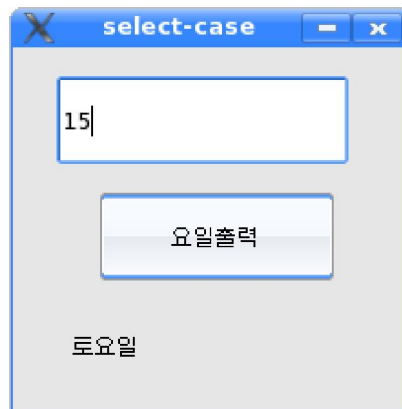
<속성>

조종체이름	속성이름	속성값
Label1	Name	lbl
	Text	콤퓨터를 배우자!
RadioButton1	Name	rdbmagen
	Text	분홍색
RadioButton2	Name	rdbyel
	Text	노란색
RadioButton3	Name	rdboran
	Text	감색

<코드>

```
PUBLIC SUB rdbmagen_Click()
lbl.Forecolor=Color.Magenta
END
PUBLIC SUB rdbyel_Click()
lbl.Forecolor=Color.Yellow
END
PUBLIC SUB rdboran_Click()
lbl.Forecolor=Color.Orange
```

END

3 . 2012년 9월 1일이 토요일이라는것을 알고 이달의 임의의 날자를 입력하면 요일을 출력하는 프로그람을 작성하여라.

<대면부>

<문제에 대한 설명>

달력을 보아라. 요일은 7을 주기로 규칙적으로 변한다.

7, 14, 21, 28일은 어떤 요일인가?

8, 15, 22, 29일은 어떤 요일인가?

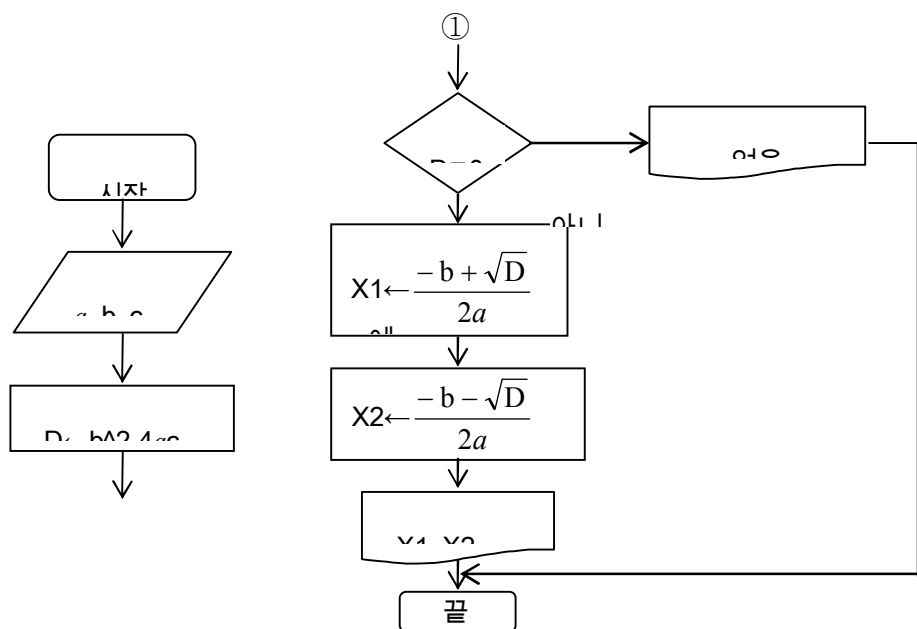
이런 식으로 보면 한개 달안에서만 보아도 8월안에서 7의 배수로 된 날들은 모두 금요일이고 그보다 1 더 큰 수에 해당한 날들은 모두 토요일이며 2 더 큰 수에 해당한 날들은 모두 일요일, 이런 식으로 되여있으므로 날자를 7로 나누어보고 나머지가 0인 날들은 금요일, 1인 날들은 토요일 등으로 변한다는것을 알수 있다.

따라서 프로그람에서는 날자를 7로 나머지연산한 결과와 0부터 6까지의 값중에서 일치한 값에 대응한 요일을 선택하여 출력하면 된다.

<코드>

```
PUBLIC SUB btnday_Click()
DIM d AS Integer
d=Val(txtday.Text)
SELECT CASE d MOD 7
CASE 0
```

```
lblday.Text="금요일"
CASE  1
lblday.Text="토요일"
CASE  2
lblday.Text="일요일"
CASE  3
lblday.Text="월요일"
CASE  4
lblday.Text="화요일"
CASE  5
lblday.Text="수요일"
CASE 6
lblday.Text="목요일"
ENDSELECT
END
```

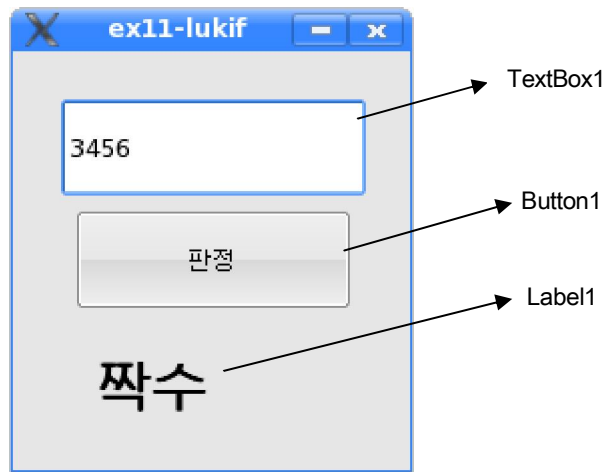
4 . 2차방정식 ax^2+bx+c=0의 실수풀이를 구하는 프로그램을 작성하여라.

①

〈코드〉

```
PUBLIC SUB btncalc_Click()
INPUT a, b, c
D=b^2-4*a*c
IF D>=0 THEN
X1=(-b+SQR(D))/(2*a)
X2=(-b-SQR(D))/(2*a)
PRINT "x1="; X1, "x2="; X2
ELSE
PRINT "없음"
ENDIF
END
```

5 . 자연수를 입력하고 단추를 찰칵할 때 짝수인가 홀수인가를 판정하는 프로그람을 작성하여라.

<대면부>

<속성>

조종체이름	속성이름	속성값
TextBox1	Name	txt1
	Text	
Button1	Name	btn1
	Text	판정
Label1	Name	lbl1

	Text	

<코드>

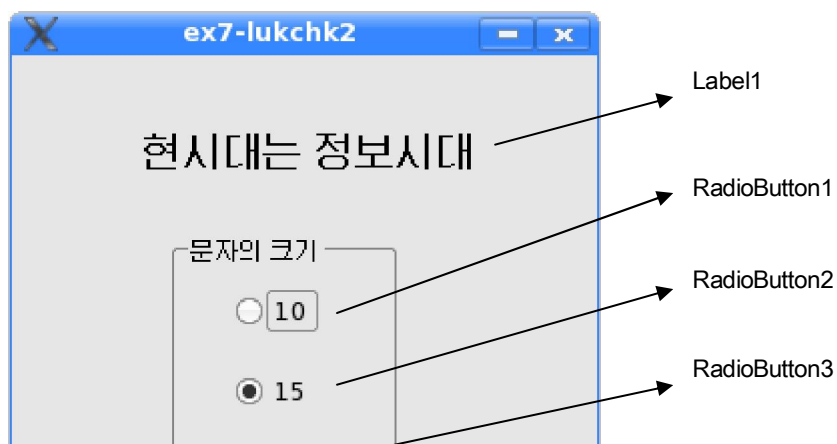
```
PUBLIC SUB btn1_Click()

DIM i AS Integer

i=Val(txt1.Text)

IF i MOD 2=0 THEN

lbl1.Text="짝수"

ELSE

lbl1.Text="홀수"

ENDIF

END
```

<코드설명>

본문칸 txt1에 입력된 자료를 수값으로 변환하여 변수 i에 넣는다. 변수 i의 값을 2로 나머지연산한 결과가 0이면(i가 짝수이면) 표식자 lbl1에 《짝수》라고 출력한다. 그렇지 않으면 즉 결과가 1이면(i가 홀수이면) 표식자에 《홀수》라고 출력한다.

6. 문자크기를 나타내는 칸안에서 문자크기를 선택하면 《현시대는 정보시대》라는 문자렬의 문자크기가 변화되게 하여라.

<대면부>

<속성>

조종체이름	속성이름	속성값
Label1	Name	lbl
	Text	현시대는 정보시대
RadioButton1	Name	rdb10
	Text	10
RadioButton2	Name	rdb15

	Text	15
RadioButton3	Name	rdb20
	Text	20

<코드>

```
PUBLIC SUB rdb10_Click()
lbl.Font.size=10
END
PUBLIC SUB rdb15_Click()
lbl.Font.size=15
END
PUBLIC SUB rdb20_Click()
lbl.Font.size=20
END
```

<코드설명>

문자크기 10을 나타내는 선택단추 rdb10을 찰칵하면 표식자의 문자크기를 10으로 변화시키기 위하여 표식자 lbl의 속성인 Font객체의 size속성을 10으로 설정한다.

15와 20인 경우도 마찬가지로 설정한다.

련습문제

1. x의 값을 입력하여 다음 식의 값을 계산하는 프로그램을 작성하여라.

$$Y=\begin{cases} x+1, & x>1 \\ x/12, & 0\leqq x\leqq 1 \\ |x|, & x<0 \end{cases}$$

2. 2012년 9월 1일이 토요일이라는것을 알고 10월의 임의의 날자를 입력하면 요일을 출력하여라.
3. 5명 학생들의 자료를 입력하고 《다음》단추를 찰칵할 때마다 다음 학생자료를 검사칸을 리용하여 설정하게 하여라.

   《통계》지령단추를 찰칵하면 남자이면서 최우등인 학생수를 표시하는 프로그램을 작

성하여라.(검사칸을 리용하되 《최우등생인가, 남자인가?》 항목을 포함하게 하여라.)

4. 학생들의 수학, 물리, 콤퓨터성적을 입력하면 학생의 성적을 평가하는 프로그람을 작성하여라.(최우등, 우등, 보통으로 출력하여라.)

제 3 절. 성적평가

《붉은별》 조작체계의 《우리》 《글》에서 《서식/페지》 항목에 있는 《테두리》 표쪽에 가보자.

여기서 《격식》 목록에서는 테두리선의 두께를 규정하게 되여있다.

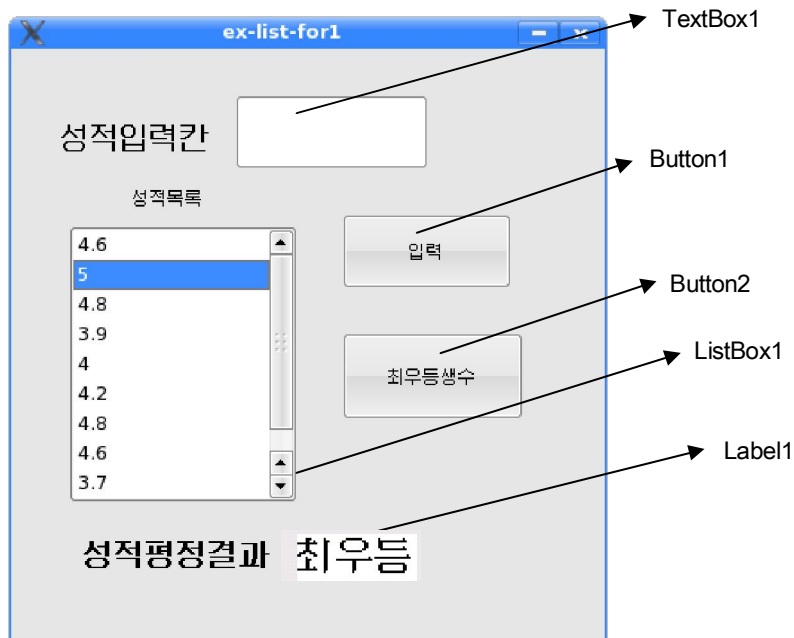
이 격식칸에는 두께를 나타내는 많은 수값과 선두께가 있는데 이가운데서 필요한것을 선택하여 설정하도록 한다.

이 절에서는 이렇게 많은 항목들가운데서 선택하여 리용할수 있는 목록칸조종체에 대하여 학습하게 된다.

자료를 반복처리해야 할 경우에 리용하는것이 순환명령문이며 많은 자료를 한눈에 보면서 선택하여 리용할수 있게 하는것이 목록칸조종체이다.

목록칸에 등록된 성적들가운데서 사용자가 선택한 성적이 최우등인가, 우등인가를 분류하고 단추를 찰칵하면 최우등생수를 출력하는 프로그람을 작성하자.

성적입력은 본문칸으로 한다.

< 속성>

조종체이름	속성이름	속성값
ListBox1	Name	lstmark
Label1	Name	lblresult
	Text	
TextBox1	Name	txtinput
	Text	
Button1	Name	btninput
	Text	입력
Button2	Name	btncalc

	Text	최우등생수

<코드>

```
PUBLIC SUB btninput_Click()
lstmark.Add(txtinput.Text)        ①
txtinput.Text = ""                ②
END
PUBLIC SUB lstmark_Click()
DIM mark AS Float
mark=Val(lstmark.Text)            ①
IF mark>=4.5 THEN                 ②
lblresult.Text="최우등"
ELSE IF mark>=3.5 THEN
lblresult.Text="우등"
ENDIF
```

```
END
PUBLIC SUB btncalc_Click()
DIM i AS Integer, k AS Integer              ①
FOR i = 0 TO lstmark.count-1                 ②
IF Val(lstmark[i].Text)>=4.5 THEN k=k+1     ③
NEXT
lblresult.Text=k
END
```

<코드설명>

btninput_Click()사건의 첫 행은 본문칸에 입력된 기호렬을 목록칸에 추가하는 코드이다. 다음행은 본문칸의 자료가 추가되였으므로 다음자료입력을 위하여 본문칸을 초기화하는 코드이다.

lstmark_Click()사건에서 mark변수는 성적을 받기 위한 실수형변수이다.

첫 행은 목록칸에서 선택된 성적을 수값으로 변환하여 변수에 값주기하는 코드이다.

둘째 행부터는 성적자료를 판정하여 4.5이상이면 표식자에 《최우등》이라고 출력하며 아니면 다시 판정하여 3.5이상이면 《우등》이라고 출력하기 위한 코드이다.

btncalc_Click()사건에서 첫 행은 순환변수와 최우등생수를 기억하기 위한 변수소개문이다.

둘째 행은 순환의 머리부인데 순환변수는 목록칸의 시작번호 0부터 마지막번호 lstmark.count–1까지 값주기된다.

셋째 행에서는 성적이 4.5이상이면 변수 k값을 하나씩 증가시킴으로써 최우등인 학생들의 인원수를 센다.

순환이 끝나면 변수 k에는 최우등생수가 기억된다. 따라서 마지막행에서는 이것을 표식자에 값주기하여 출력하게 하였다.

실례에서 보는것처럼 목록칸은 많은 값들을 사용자가 보면서 필요한것을 선택하여 리용하는데 편리한 조종체이며 순환명령문은 같은 처리를 여러번 반복하게 하는데 편리한 명령문이다.

1. 순환명령문

단 몇개의 명령을 가지고 많은 반복처리를 하도록 하는것이 순환명령문이다.

여기서는 무조건순환명령문 FOR~NEXT와 조건순환명령문 DO [WHILE]~LOOP에 대하여 보기로 한다.

1) 무조건순환명령문 FOR~NEXT명령문

<형식>

```
FOR 변수=시작값 TO 끝값 [STEP 걸음값]
명령문들
NEXT
```

여기서 FOR, TO, STEP, NEXT는 《…하는 동안》, 《…에로》, 《걸음》, 《다음》이라는 뜻이다.

이 명령문은 변수를 시작값부터 끝값까지 걸음값만큼씩 변화시키면서 명령문들을 반복수행한다.

먼저 변수에 시작값을 넣고 순환부안의 명령문들을 수행하고 걸음값만큼 변수값을 증가(감소)시킨다. 다음 변수값을 끝값과 비교하여 끝값을 넘지 않았으면 순환부분을 반복하고 매번 변수값을 걸음값만큼 증가(감소)시킨다.

순환은 끝값을 넘으면 끝난다.

STEP부분이 생략된 경우 걸음값을 1로 하여 순환한다. 여기서 변수는 수값형(Byte, Short, Integer, Float)이여야 하며 반드시 사건수속안에서 DIM으로 선언하여야 한다.

주의할 점은 걸음값이 정수인 경우는 끝값이 시작값보다 커야 하며 걸음값이 부수인 경우에는 시작값이 끝값보다 커야 한다. 그렇지 못한 경우는 코드가 전혀 실행되지 않는다.

NEXT는 FOR와 개수가 맞아야 한다. 또한 순환에서 FOR가 여러번 있는 경우에는 마지막으로 선언한 FOR에 대한 NEXT가 먼저 있어야 하며 이런 식으로 제일먼저 시작된

FOR문에 대한 NEXT가 제일 마지막에 있어야 한다.

례 1: 1부터 10까지 자연수들의 합을 구하여라.

<코드>

```
PUBLIC SUB btncalc_Click()
DIM i AS Integer, s AS Integer
FOR i=1 TO 10
s=s+i
NEXT
txtres.Text=s
END
```

<코드설명>

코드에서 i는 순환변수이다. 변수 s는 선언부가 실행될 때 0으로 초기화된다.

순환변수가 1을 가지고 순환부안에 들어오면 0을 가진 변수 s와 더하기된다. 따라서 변수 s는 1이 된다.

NEXT를 만나면 다시 FOR부분으로 올라가 순환변수값이 2가 된다. 다시 순환부안에서 본래 있던 s의 값 1에 2가 더해져 1+2의 결과가 변수 s에 들어간다. 다시 NEXT를 만나면 FOR명령문행에 올라가 이번에는 순환변수에 3을 넣고 순환부를 집행한다. 이런 식으로 순환변수가 10이 될 때까지 순환부의 명령문이 집행되면 변수 s에는 1부터 10까지 더한 합이 들어가있게 된다.

다음행에서 본문칸에 합을 출력한다.

례 2: 구구표의 결과들을 출력한다고 하자. 그러면 무조건순환명령문을 어떻게 리용하여 할수 있겠는가?

다음의 코드를 통하여 보기로 하자.

```
FOR i=1 to 9
FOR j=1 to 9
PRINT i*j
Next
Next
```

이것이 구구표결과를 출력하는 프로그람이다.

구구표에는 1계단일 때 1부터 9까지를 차례로 1에 곱하여 결과를 구하고 2계단일 때 1부터 9까지를 차례로 2에 곱하여 결과를 구하며 마찬가지로 9계단까지를 반복한다. 따라서 계단수를 하나 고정시키면 그에 대하여 1부터 9까지를 차례로 곱하여 계산하게 되여야 한다. 그러므로 계단수를 바깥순환에 놓고 한개 값이 정해지면 그에 대하여 순환부안에서 다시 곱하는 수를 취해주는 안순환이 또 있어야 한다.

이 코드에서는 바깥순환의 한개 값에 대하여 안순환이 처음값부터 마지막값까지 순환되여야 바깥순환의 다음 값으로 넘어가게 된다. 이러한 순환을 2중순환이라고 하며 이런 순환이 계속되는 순환을 다중순환이라고 한다.

2) 조건순환명령문 DO [WHILE]~LOOP

형식:

```
DO [WHILE 조건식]
…
```

LOOP

기능: 조건식을 만족하거나 순환안에서 EXIT를 만날 때까지 순환을 반복한다.

순환되는 명령문들은 DO와 LOOP사이에 있는 명령문들이다.

만일 WHILE부분이 생략된 경우에는 무한순환이 계속되거나 일부 조건이 탈퇴조건을 만날 때까지 순환이 계속된다.

생략되지 않은 경우에는 조건식을 검사하여 결과가 True인 동안 반복하다가 False가 되였을 때 순환이 중지된다.

순환이 시작될 때 초기에 조건식값이 False이면 순환은 절대로 일어나지 않는다.

례 1: S=1+2+3+…의 합이 10 000을 넘을 때까지의 합과 그때의 수자를 구하는 프로그람을 작성하여라.

```
PUBLIC SUB btnresult_Click()
DIM s AS Long, i AS Integer
DO WHILE s<=10000
i=i+1
s=s+i
LOOP
PRINT s, i
END
```

례 2: 실행이 시작되여 10초가 될 때까지 계수되는 초수를 실수로 출력하는 프로그람을

작성하여라.

```
PUBLIC SUB Form_Open()
DO WHILE Timer<=10
PRINT Timer
LOOP
END
```

여기서 Timer는 프로그람의 시작으로부터 지나간 초수를 실수로 돌려주는 함수이다.

2. 목록칸조종체

목록칸조종체(ListBox) 는 선택할수 있는 여러개의 본문항목들을 표시하는 조종체이다.

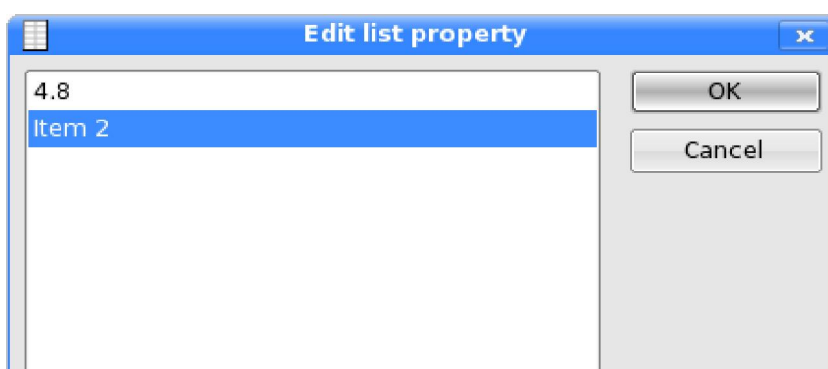
사용자는 이 본문항목들가운데서 필요한 하나 또는 그이상의 항목들을 선택하여 처리할수 있다.

목록칸에 본문항목들의 등록은 설계시에도 할수 있고 실행시에도 할수 있다.

설계시 목록칸에 항목들을 등록하는 방법은 다음과 같다 .

도구칸에서 ListBox조종체를 선택하여 Form에 배치한다.

속성칸에서 List속성을 찾아 그의 오른쪽에 있는 입력칸의 ...을 찰칵한다.

목록속성편집대화칸(Edit list property)이 펼쳐진다.

Insert단추를 찰칵한다. 그러면 아래의 본문칸에 Item1이라는 항목이름이 나타난다.

본문칸의 Item1대신에 4.8을 입력한다.

Insert단추를 찰칵한다.

본문칸에 표시된 Item2항목이름대신 5를 입력한다.

같은 방법으로 셋째, 넷째 항목들을 입력한다.

입력이 끝나면 목록에 입력한 항목들이 차례로 정확히 등록되였는가를 확인하고 OK단추를 찰칵한다. 그러면 실행시 항목들이 나타나는것을 볼수 있다.

등록된 항목들의 삭제는 항목을 선택하고 Delete단추를 찰칵하는 방법으로 진행한다.

실행시 목록칸에 항목들을 등록하는 방법은 다음과 같다 .

코드를 작성하여 진행한다.

3명 학생의 성적이 실행시에 목록칸에 등록되게 하려면 다음과 같이 코드작성을 할수 있다.

```
PUBLIC SUB Form_Open()   Form이 펼쳐질 때 발생하는 사건
ListBox1.Add(“4.8” ,1)      목록칸의 1번항목으로 추가
ListBox1.Add(“5”, 2)        목록칸의 2번항목으로 추가
```

ListBox1.Add("4.6", 3)   목록칸의 3번항목으로 추가
END

코드실례에서 보는것처럼 실행시 목록칸에 항목들을 등록하려면 목록칸조종체가 가지는 속성들과 방법들을 알아야 한다.

- 속성

Count: 목록에 등록된 항목의 개수를 돌려주는 속성이다.
Text: 목록칸에서 선택된 항목의 내용 또는 지적한 항목의 내용을 돌려준다.
실행시 대면부에서 선택한 항목의 내용:

객체.Text

코드에서 번호를 지적한 항목의 내용:

객체[항목번호].Text

객체는 목록칸객체이며 항목번호는 목록칸안에서 항목의 번호인데 0번부터 시작된다.
례 : PRINT Lstnum.Text
이 코드가 실행되면 목록칸에서 사용자가 선택한 항목의 내용이 조작탁창문에 출력된다.
목록칸의 i번째 항목의 내용은 ListBox1[i].Text로 규정된다.
례: PRINT Lstnum[1].Text
이 코드가 실행되면 목록칸의 둘째 항목의 내용이 출력된다.
List: 목록칸에 등록되여있는 전체 항목들을 문자렬로 주는 속성이다. 이때 매개 항목은 행바꾸기기호로 구별되여있다.

례:
```
PUBLIC SUB Form_Open()

PRINT ListBox1.list

END
```

이 코드가 실행되면 조작탁창문에 목록칸의 항목이 항목별로 서로 다른 행에 순서대로 나가면서 마지막까지 출력된다.

Index: 목록안에서 선택된 항목의 번호이다. 선택되지 않았으면 -1이다.

- 방법

Add: 이 방법은 목록칸의 지적된 첨수위치에 기호렬로 된 항목을 추가하는 방법이다.

형식: 목록칸조종체이름.Add(항목 [,첨수])

항목은 기호렬형이다. 만일 첨수가 생략되면 목록칸의 제일 마지막에 추가한다.

례: ListBox1.Add(“김영철”, 1)

이것은 목록칸의 두번째 항목으로 《김영철》이라는 문자렬을 추가한다.

우리가 설정한 문제에서는 목록칸에 1부터 50까지의 자연수들이 등록되여있어야 하므로 실행시에 등록되게 하려면 다음과 같이 할수 있다.

```
List1.Add(1)
List1.Add(2)
…
List1.Add(50)
```

Remove: 이 방법은 첨수로 지적된 항목을 목록칸에서 삭제한다.

형식: 목록칸조종체이름.Remove(첨수)

례: List1.Remove(2)

이것은 목록칸의 세번째 항목을 제거하는 코드행이다.

Clear: 목록칸안의 내용을 전부 지운다.

형식: 목록칸조종체이름.Clear

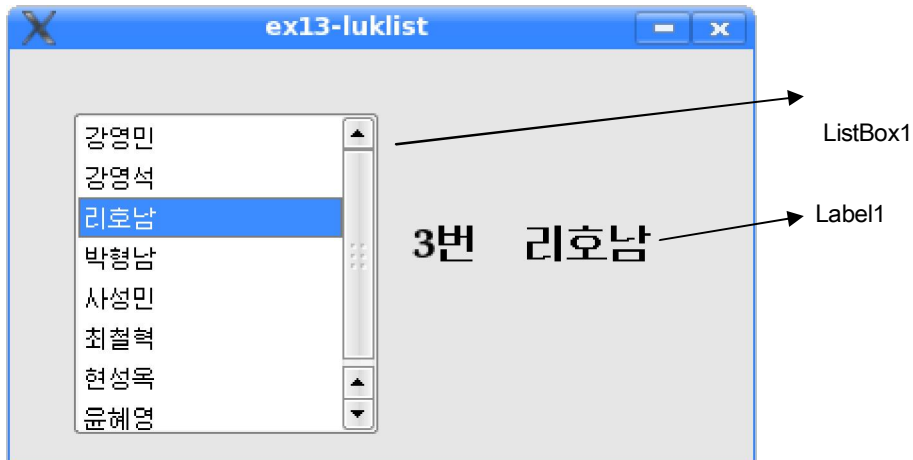
SetFocus 사건

조종체에 초점을 주려고 할 때 호출한다.

실 습

1 . 목록칸에서 학생의 이름을 선택하면 그의 이름과 출석번호를 출력하는 프로그람을 작성하여라.

<대면부>

<속성>

조종체이름	속성이름	속성값
ListBox1	Name	lst1

Label1	Name	lblnum
	Text	

<코드>

```
PUBLIC SUB lst1_Click()
lblname.Text =lst1.Text                    ①
lblnum.Text =Str(lst1.Index + 1) & "번"    ②
END
```

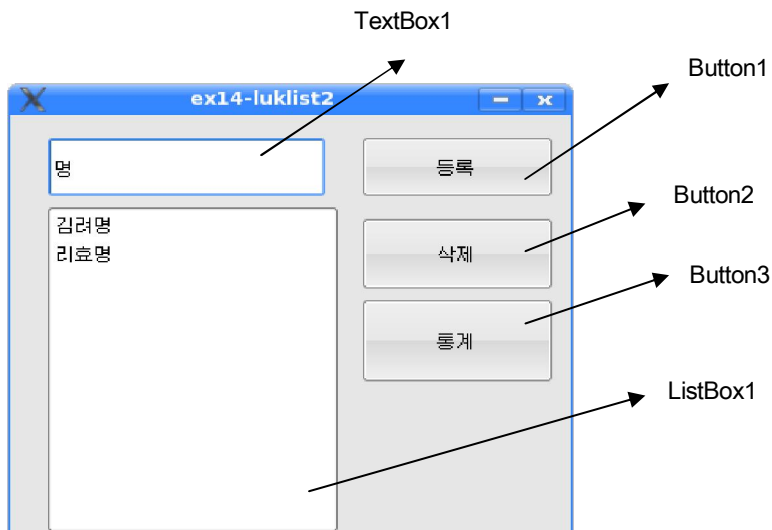
<코드설명>

목록칸에서 항목을 선택하면 Click사건이 발생하므로 이 사건수속에 코드를 작성하였다. 사건수속의 ①행은 선택된 항목내용(lst1.Text)을 이름을 표시하는 표식자(lbln ame)에 표시하기 위해 그의 Text속성에 값주기를 하였다.

②행에서 lst1.Index는 목록칸안에서 선택된 항목의 첨수번호인데 출석번호는 1번부터 시작되므로 1을 더하였다.(목록칸의 첨수번호는 0부터 시작한다.)

2. 학생의 이름을 입력하고 《등록》단추를 찰칵하면 목록칸에 이름을 등록하고 목록칸에서 이름을 선택하고 《삭제》단추를 찰칵하면 이름을 목록에서 삭제하며 《통계》단추를 찰칵하면 학생의 인원을 출력하는 프로그람을 작성하여라.

<대면부>

<속성>

조종체이름	속성이름	속성값
TextBox1	Name	txtname
Button1	Name	btnadd
	Text	등록
Button2	Name	btnremove
	Text	삭제
Button3	Name	Btncalc
	Text	통계
ListBox1	Name	lstname

<코드>

```
PUBLIC SUB btnadd_Click()
lstname.Add(txtname.Text)
txtname.Text = ""
END

PUBLIC SUB btnremove_Click()
lstname.Remove(lstname.Index)
END

PUBLIC SUB btncalc_Click()
txtname.Text=lstname.Count
END
```

<코드설명>

btnadd_Click사건은 《등록》단추를 찰칵하였을 때 발생하는 사건에 대한 수속이다.

본문칸에 입력한 이름은 txtname의 Text속성에 반영된다. 그러므로 사건수속의 첫 행에서는 이름을 등록하기 위하여 목록칸lstname의 Add방법을 리용하여 항목내용 txt name.Text를 추가하도록 한다. 추가함과 동시에 본문칸의 내용은 빈 문자렬로 만든다.

btnremove_Click사건은 《삭제》단추를 찰칵하였을 때 발생하는 사건에 대한 수속이다.

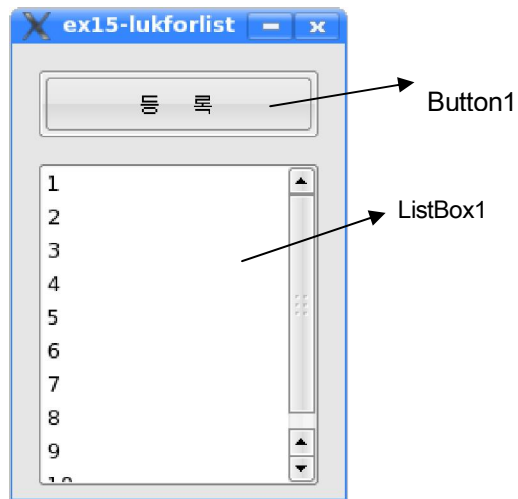
사건수속에서 Lstname.Index는 목록칸에서 선택한 항목의 첨수이다.

Lstname객체의 Remove방법의 첨수자리에 lstname.Index를 넣음으로써 선택한 항목을 제거하게 된다.

《통계》단추를 찰칵하면 btncalc_Click사건수속에 있는 명령문이 집행된다. 여기서 lstname.Count는 목록칸에 있는 항목의 개수이다. 이것을 문자렬로 변환하여 본문칸에 출력하게 된다.

3 . 지령단추를 찰칵하면 목록칸에 1부터 50까지의 자연수가 단번에 등록되게 하여라.

<대면부>

<속성>

조종체이름	속성이름	속성값
Button1	Name	btnadd
	Text	등록
ListBox1	Name	lstnum

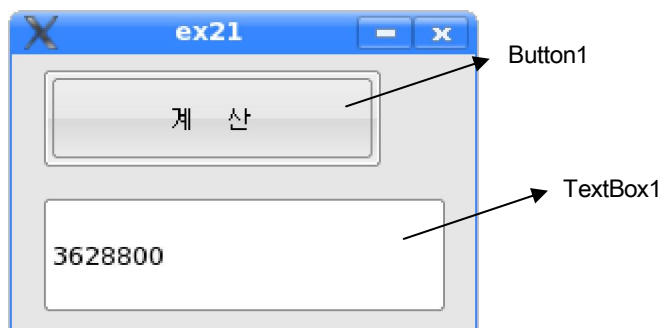
<코드>

```
PUBLIC SUB btnadd_Click()

DIM i AS Integer

FOR i=1 TO 50

lstnum.Add(i)

NEXT

END
```

<코드설명>

목록칸의 Add방법의 인수로 1부터 50까지의 수들을 하나씩 순서대로 넣으면 50개의 수가 등록될수 있다. 따라서 순환명령문을 리용하였다.

4 . 10차례곱(1부터 10까지 자연수들의 적)을 구하여라.

<대면부>

<속성>

조종체이름	속성이름	속성값
Button1	Name	btncalc
	Text	계산
TextBox1	Name	txtres

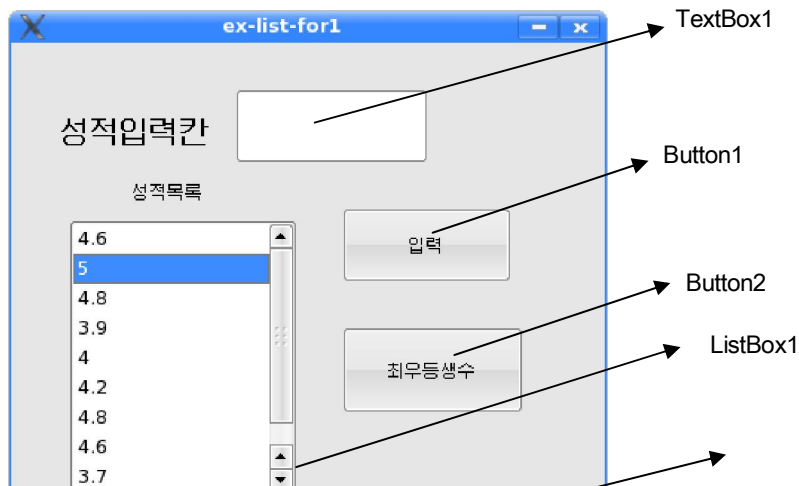
<코드>

```
PUBLIC SUB btncalc_Click()
DIM i AS Integer, y AS Float
y=1
FOR i=1 TO 10
y=y*i
NEXT
txtres.Text=y
END
```

<코드설명>

알아야 할것은 적을 기억할 변수가 선언부에서 0으로 초기화된 조건에서 여기에 순환 변수를 곱하면 아무리 곱해도 0만 나오게 된다. 따라서 곱하기하는 순환이 시작되기 전에 적을 기억할 변수에 1을 값주기하였다.

5. 목록칸에 등록된 성적들가운데서 사용자가 선택한 성적이 최우등인가, 우등인가를 분류하는 프로그람을 작성하여라.

단추를 찰칵하면 최우등생수도 출력하여라.

Label1

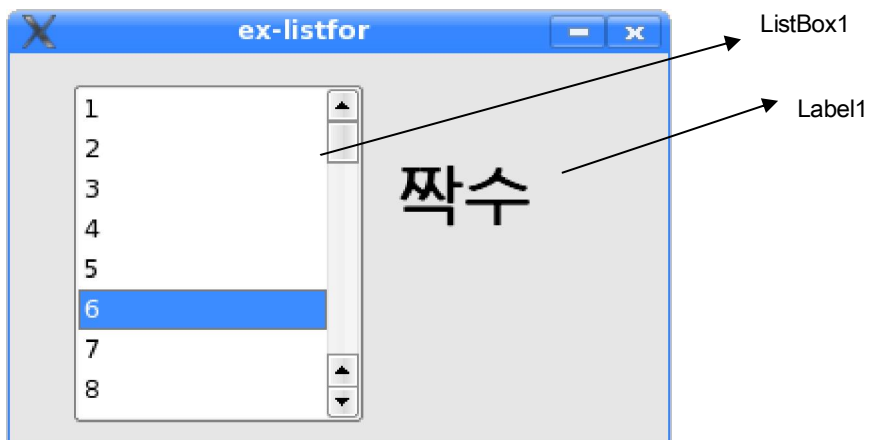
<코드>

```
PUBLIC SUB btninput_Click()
lstmark.Add(txtinput.Text)
txtinput.Text=""
END
PUBLIC  SUB  lstmark_Click()
DIM  mark  AS  Float
mark=Val(lstmark.Text)
IF  mark>=4.5  THEN
lblresult.Text="최우등"
ELSE  IF  mark>=3.5  THEN
lblresult.Text="우등"
```

```
ENDIF
END
PUBLIC SUB btncalc_Click()
DIM i AS Integer, k AS Integer
FOR i=0 TO lstmark.Count - 1
IF Val(lstmark[i].Text)>=4.5 THEN k=k + 1
NEXT
lblresult.Text=k
END
```

6 . 목록칸에 등록된 1부터 50까지 자연수가운데서 사용자가 선택한 수가 짝수인가 홀수인가를 판정하는 프로그람을 작성하여라.

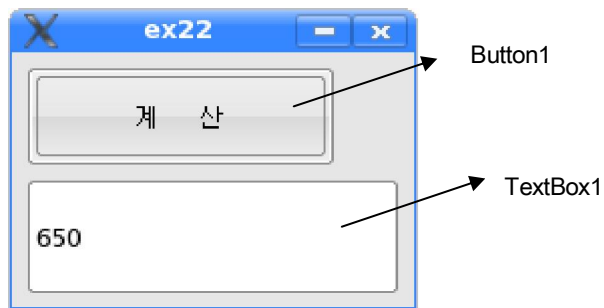
<대면부>

<속성>

조종체이름	속성이름	속성값
ListBox1	Name	lstnum
Label1	Name	lblresult

<코드>

```
PUBLIC SUB lstnum_Click()
DIM number AS Integer
number=Val(lstnum.Text)
IF number MOD 2=0 THEN
lblresult.Text="짝수"
ELSE
lblresult.Text="홀수"
ENDIF
END

PUBLIC SUB Form_Open()
DIM i AS Integer
FOR i=1 TO 50
lstnum.Add(i)
NEXT
END
```

7 . 1부터 50까지 자연수들가운데서 짝수들의 합을 구하여라.

<대면부>

<속성설정>

조종체이름	속성이름	속성값
Button1	Name	btncalc
	Text	계산
TextBox1	Name	txtres

<코드>

```
PUBLIC SUB btncalc_Click()
DIM i AS Integer, s AS Integer
FOR i=2 TO 50 STEP 2
s=s+i
NEXT
txtres.Text=s
```

END

<코드설명>

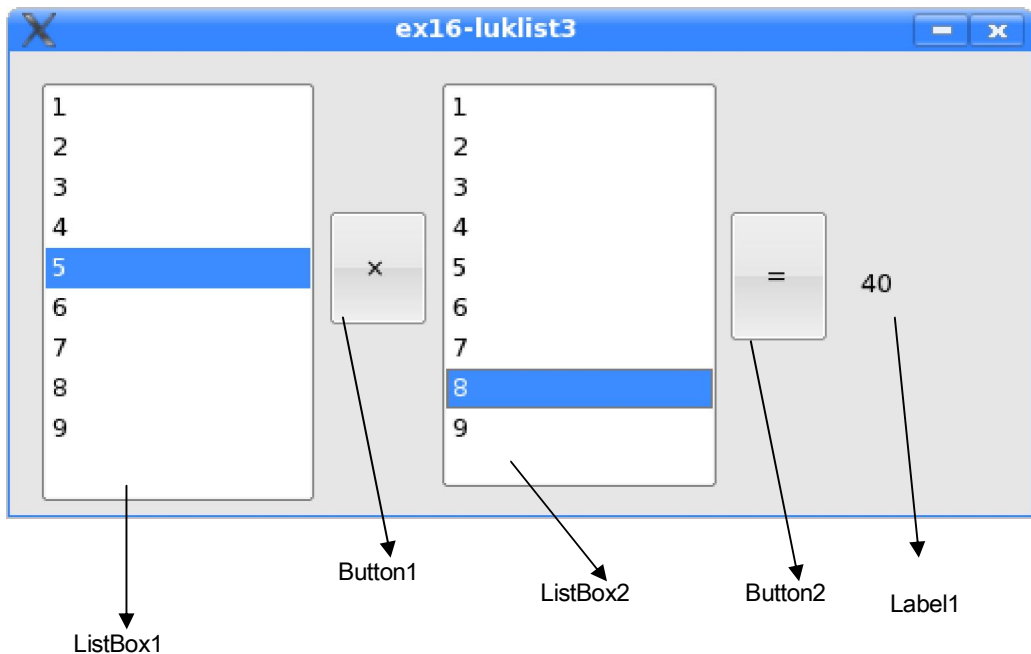
이 문제는 례 1에서처럼 합을 구하는 문제인데 짝수들의 합을 구하여야 한다.

따라서 순환변수의 값이 짝수값들만을 가져야 하므로 순환시작은 0부터 하며 걸음값을 2로 주면 순환변수 i는 2, 4, 6, …, 50과 같은 값을 가지고 순환부에 들어와 처리된다.

만일 순환변수 i를 1부터 50까지 걸음값 1로 순환시키려면 순환부안에서 i값이 짝수인가를 판정하여 조건을 만족하는 수값만 더하기해야 하므로 사건수속의 3번째 행에서 i값을 판정하여 짝수(2로 나눈 나머지가 0이면)일 때마다 더하기명령을 집행하도록 하였다.

8. 두개의 목록칸에서 1부터 9사이의 임의의 수값을 선택하고 =단추를 찰칵하면 두수를 곱한 적을 구구표형식으로 출력하여라.

<대면부>

<속성>

조종체이름	속성이름	속성값
ListBox1	Name	lst1
	Name	lst2
Button1	Name	btn1
	Text	×
Button2	Name	btn2
	Text	=
Label1	Name	lbl1

<코드>

```
PUBLIC SUB Form_Open()

DIM i AS Integer

FOR i=1 TO 9

lst1.Add(i)

lst2.Add(i)
```

```
NEXT
END

PUBLIC SUB btn2_Click()
lbl1.Text=Str(Val(lst1.Text)*Val(lst2.Text))      ①
END
```

<코드설명>

=를 표시하고있는 단추 btn2_Click사건수속의 ①행에서 lst1.Text는 첫 목록칸에서 선택한 항목의 내용이다. 역시 lst2.Text는 둘째 목록칸에서 선택한 항목의 내용이다.

이 두 속성에 Val함수를 적용하였으므로 기호렬이 수값으로 변환되여 곱하기된다. 즉 첫 목록칸에서 선택한 수값과 둘째 목록칸에서 선택한 수값이 곱하기된다.

이것을 Str함수를 리용하여 기호렬로 변환하여 표식자 lbl1.Text에 값주기함으로써 곱하기된 결과는 기호렬로 표식자에 출력된다.

련습문제

1. 1, 2, 3, 5, 8, 13 …과 같은 수렬이 있다. 수렬의 합이 100만을 넘을 때까지 더한 수의 개수와 합을 구하는 프로그람을 작성하여라.
2. 어떤 자동차가 V_0의 속도로 떠나 a의 가속도로 등가속직선운동을 하는데 t시간동안에 간 거리를 1분간격으로 목록칸에 출력하여라.

제 4 절. 도형편집기

프로그람의 시각적인 효과를 높이는데서 도형처리가 차지하는 몫은 매우 크다.

도형사용자대면부설계에서는 조종체들의 배렬을 직관적으로 하는것과 함께 색조화를 보기 좋게 실현하고 적당한 위치에 그림을 잘 안받침하면 보다 훌륭한 효과를 거둘수 있다 . 또한 프로그람작성에서는 식이나 문자렬의 연산처리뿐만이 아니라 그라프작성이나 도형그리기 , 마우스조종과 같은 도형처리과제가 많이 제기된다 .

Gambas에서는 도형처리를 위해 조종체와 함께 클라스를 사용한다 .

간단한 도형편집기를 작성해보는 과정을 통하여 Gambas에서의 도형처리수법에 대하여 익히도록 한다 .

대면부를 다음과 같이 구성한다.

<속성>

조종체이름	속 성	값
DrawingArea	Stretch	True
Button1	Text	선분그리기
Button2	Text	타원그리기

<코드>

```
PUBLIC SUB Button1_Click()
DIM x, y, a, b AS integer
x=100
y=100
a=100
b=200
Draw.Begin(DrawingArea1)
Draw.Forecolor=Color.Red
Draw.Line(x, y, a, b)
Draw.End
END
PUBLIC SUB Button2_Click()
DIM x, y, w, h AS integer
x=100
y=100
w=100
h=200
Draw.Begin(DrawingArea1)
Draw.Forecolor=Color.Green
Draw.Ellipse(x, y, w, h)
Draw.End
END
```

<코드설명>

Draw.Begin(DrawingArea1): 그리기시작을 선언하는 명령문이다.

Gambas에서 기본도형처리는 DrawingArea조종체와 Draw클라스를 리용하여 진행하는데 그림그리기를 시작하기 전에 Begin이라는 방법을 먼저 써주고 도형처리를 진행하여야 한다.

이 방법의 형식은 다음과 같다.

Draw.Begin(그리기하려는 조종체의 이름)

Draw.Forecolor=Color.Green: 그리려는 도형의 전경색을 설정하는 명령문이다.

Draw.Ellipse(x, y ,w, h): 타원을 그리기하는 방법이다. 이 방법은 점 (x, y)를 시작점으로 하고 w만 한 너비와 h만 한 높이에 해당한 직4각형에 내접하는 타원을 그린다.

Draw.End: 그리기의 끝을 알리는 명령문이다.

도형처리마감에는 End방법을 반드시 써주어야 한다.

이 방법의 형식은 다음과 같다.

Draw.End

1. 색

일반적으로 도형그리기에서는 전경색과 배경색을 리용한다.

흰종이우에 붉은색으로 그림을 그릴 때 전경색은 붉은색이고 배경색은 흰색이다.

도형그리기를 할 때 전경색을 규정하여주면 그 색으로 도형이 그려지며 다시 전경색을 다른색으로 설정하여주면 그 색으로 그려진다.

배경색도 이와 마찬가지이다.

전경색이나 배경색을 특별히 설정하지 않으면 본래의 배경색과 그리는 색이 배경색과 전경색으로 리용된다.

Gambas에서 색설정은 색상수(Color Constants)에 의하여 진행한다.

색설정은 다음과 같은 방법으로 진행할수 있다.

- 체계색상수의 리용

Gambas에서 많이 쓰이는 색값들을 색상수로 따로 정의하여 주었다.

기본색들을 보면 다음과 같다.

체계상수	수값	설명
Black	0	검은색
Blue	255	푸른색
Green	65 280	풀색
Cyan	65 535	하늘색
Red	16 711 680	붉은색
Magenta	16 711 935	분홍색
Yellow	16 776 960	노란색

White	16 777 215	흰색
Gray	8 421 504	회색
Orange	16 744 448	감색

여기에서 수값들은 해당한 색에 대응하는 수값들이다.

결국 색을 지정할 때 체계상수를 쓸수도 있고 거기에 대응하는 수값을 쓸수도 있다.

```
PUBLIC SUB Button1_Click( )
Button1.Background=Color.Blue
END
```

이 프로그람을 실행하면 지령단추의 배경색이 푸른색으로 된다.

색상수를 리용하여 색을 지정하는 형식은 다음과 같다.

형식: Color.색상수

• RGB방법의 리용

RGB방법은 붉은색, 풀색, 푸른색을 적당하게 혼합하여 필요한 색을 얻게 하는 색설정방법이다.

이 방법의 형식은 다음과 같다.

형식: Color.RGB(붉은색, 풀색, 푸른색)

여기에서 매 색갈의 값은 0~255범위의 값을 가질수 있다.

례 :
```
PUBLIC SUB Button1_Click( )
  Button1.Background=Color.RGB(255, 0, 0)
  END
```

이 프로그람을 실행하면 지령단추의 배경색이 붉은색으로 된다.

만일 Button1.Background=Color.RGB(0, 255, 0)와 같이 써주면 도형의 배경색은 풀색으로 설정되며 Button1.Background=Color.RGB(0, 0, 255)로 써주면 도형의 배경색은 푸른색으로 설정된다. 또한 Button1.Background=Color.RGB(100, 50, 200)으로써 주면 도형의 배경색은 붉은색이 100, 풀색이 50, 푸른색이 200만큼 섞인 색으로 된다. 즉 합성된 색갈이 나타난다.

색갈값이 0이면 색갈의 세기가 제일 작고 255이면 세기가 제일 크다 . 그러므로 RGB방법을 리용하면 사용자들이 필요한 색을 설정할수 있다.

2. 그림그리기조종체

그림그리기조종체(DrawingArea) 에서는 프로그람작성자의 요구대로 그림을 그리거나 도형처리를 할수 있다.

- 속성

Name속성은 Gambas의 코드작성시 조종체의 이름속성이다.

Cached속성은 그리기구역의 내용이 기억기에 고속완충된다는것을 지적하여준다.

그림그리기조종체에서 그리기조작을 하기 위해서는 Cached속성을 True로 설정해주어야 한다.

Width, Height속성은 그림그리기조종체의 너비와 높이의 값을 보여주는 속성이다.
대면부설계시 이 값을 설정할수도 있고 코드로 이 값을 얻거나 설정할수 있다.

```
PUBLIC SUB Form_Open( )
DrawingArea1.Width=300
DrawingArea1.Height=200
END
```

우의 프로그람을 실행하면 Form이 적재될 때 DrawingArea1의 너비와 높이가 각각 300, 200으로 된다.
Font속성은 프로그람실행시 그림그리기조종체에 표시되는 문자의 형태와 크기를 결정하는 속성이다.
대면부설계시 이 값을 설정할수 있다.
코드로는 다음과 같이 서술할수 있다.

```
PUBLIC SUB Form_Open( )
DrawingArea1.Font.name="천리마"
DrawingArea1.Font.italic=True
DrawingArea1.Font.size=5
END
```

Background, Foreground는 해당 조종체의 배경색과 전경색을 변화시키는 속성이다.
대면부설계시에 이 값을 설정할수 있다.
또한 코드로서 이 값을 얻거나 설정할수 있다.

```
PUBLIC SUB Form_Open( )
DrawingArea1.Background=Color.Red
DrawingArea1.Forecolor=Color.Blue
END
```

우의 프로그람을 실행하면 Form이 적재될 때 DrawingArea1객체의 배경색은 붉은색으로 되고 거기에 그려지는 도형색은 푸른색으로 된다.

Mouse속성은 프로그람실행시 해당 조종체의 범위안에서 마우스지시자의 모양을 어떻게 변화시키겠는가를 결정하는 속성이다.

대면부설계시에 이 값을 설정할수 있다. 또한 코드로 이 값을 얻거나 설정할수 있다.

```
PUBLIC SUB Form_Open( )
DrawingArea1.Mouse=4
END
```

우의 프로그람을 실행하면 DrawingArea1에서의 마우스지시자의 모양이 4번모양으로 된다.

Border속성은 해당 조종체의 테두리선형식을 변화시키는 속성이다.

이 속성 역시 설계시와 코드작성시 설정할수 있다.

Stretch속성은 이 조종체에 그림을 적재할 때 그림이 조종체의 크기보다 크거나 작아도 조종체의 크기에 맞게 그림을 줄이거나 늘여 적재시키는 기능을 설정하는 속성이다.

- 사건

MouseMove: 마우스단추를 누른 상태에서 이동할 때 일어난다.

MouseUP: 마우스단추를 눌렀다가 놓았을 때 일어난다.

MouseDown: 마우스단추를 누를 때 일어난다.

3. Draw클라스

그림그리기조종체에서 도형처리는 Draw클라스를 리용하여 진행한다.

미술가가 그림을 그리는것과 비교하여 말하면 그림그리기조종체는 그림을 그리는 화판이고 Draw클라스는 붓, 수채화구와 같은 도구라고 볼수 있다.

그림그리기조종체에 그림을 그리려면 그리기 전에 Draw클라스의 Begin(방법)을 반드시 호출해야 한다. 또한 도형처리를 끝내려면 Draw클라스의 End(방법)를 반드시 호출해야 한다.

클라스에 대한 개념

- 속성

Fillstyle: 도형의 내부색을 어떤 형식으로 칠하겠는가를 결정하는 속성인데 수값으로 0 ~14까지의 값을 줄수 있다.

Fillcolor: 도형의 내부색을 결정하는 속성이다.

도형의 내부색은 수값 또는 R, G, B값 또는 색상수로 지정할수 있다.

FillX, FillY속성들은 다각형의 내부를 그릴 때 시작점의 자리표를 설정하거나 돌려준다.

Font: 본문표시에 리용된다.

Invert: 그리려는 도형들의 화소색을 반전하겠는가를 규정한다.

Linestyle: 도형의 테두리선을 그릴 때 선의 형태를 설정하거나 돌려준다.

Linewidth: 선의 굵기를 설정하거나 돌려준다.

Transparent: 그리기구역에 본문을 표시할 때 배경색이 나타나게 하겠는가를 지정한다.

• 방법

Begin: 그리기구역에 대한 그림그리기시작을 선포한다.

End: 그림그리기를 끝낸다.

Point: 그리기구역에 하나의 점을 찍는다.

Text: 문자렬을 쓰는데 리용한다.

문자쓰기방법의 형식은 다음과 같다.

Draw.Text(text AS string, x AS integer, y AS integer[, width AS integer, height AS integer])

이 방법은 text로 지정한 문자렬을 주어진 자리표 (x,y)의 위치에 표시한다.

Rect: 직4각형을 그린다.

직4각형그리기방법의 형식은 다음과 같다.

Draw.Rect(x AS integer, y AS integer, w AS integer, h AS integer)

이 방법은 점 (x, y)를 시작점으로 하여 w만 한 너비와 h만 한 높이에 해당한 직4각형을 그린다.

Line: 직선을 그린다.

직선그리기방법의 형식은 다음과 같다.

Draw.Line(x1 AS integer, y1 AS integer, x2 AS integer, y2 AS integer)

이 방법은 두 점 (x1, y1), (x2, y2)를 맺은 선분을 그린다.

Ellipse: 타원, 원, 부채형을 그린다.

타원그리기방법의 형식은 다음과 같다.

Draw.Ellipse(x AS integer, y AS integer, width AS integer, height AS integer[, start AS Float, length AS Float])

이 방법은 점 (x, y)를 시작점으로 하고 width, height를 각각 너비와 높이로 하는 직4각형에 내접하는 타원을 그린다.

생략가능한 start와 length를 지적해주면 start에서 시작되는 각도에서 length로 지적한 값만 한 활등의 길이로 부채형을 그리게 된다.

자리표계

도형처리를 진행하자면 해당한 객체들에서 자리표지정을 해야 하며 그러자면 자리표계에 대하여 알아야 한다.

자리표계는 조종체들에서 점들의 위치를 정의하는 2차원평면이다.

Gambas의 자리표계는 수학에서 학습한 자리표계와 차이난다.

실 습

1. 지령단추 3개를 창조하고 매 단추를 찰칵할 때마다 그 지령단추의 배경색이 각각 붉은색, 풀색, 푸른색으로 변화되게 하여라.

<대면부>

<속성>

조종체이름	속성	값
Button1	Text	붉은색
Button2	Text	풀색
Button3	Text	푸른색

<코드작성>

```
PUBLIC SUB Button1_Click( )
Button1.Backcolor=Color.Red
END
PUBLIC SUB Button2_Click( )
Button2.Backcolor=Color.Green
END
PUBLIC SUB Button3_Click( )
Button3.Backcolor=Color.Blue
END
```

2. 4개의 본문칸조종체를 창조하고 붉은색, 풀색, 푸른색의 값을 입력하여라. 그다음 지령단추를 찰칵하면 이 세가지 색갈이 혼합된 색이 네번째 본문칸조종체의 배경색이 되게 하여라.

<대면부>

<속성>

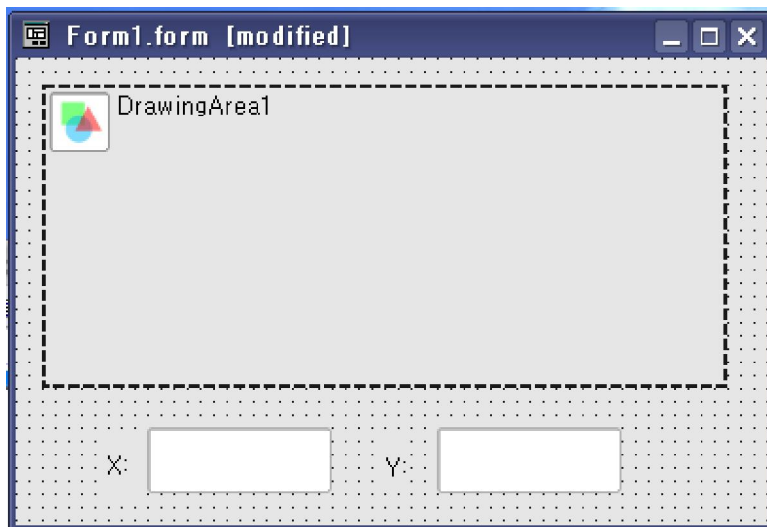
조종체이름	속성	값
TextBox1	Text	
TextBox2	Text	
TextBox3	Text	
TextBox4	Text	
Button1	Text	색합성

<코드>

```
PUBLIC r AS integer

PUBLIC g AS integer

PUBLIC b AS integer

PUBLIC SUB TextBox1_Change( )

r=TextBox1.Text

END

PUBLIC SUB TextBox2_Change( )
```

```
g=TextBox2.Text

END

PUBLIC SUB TextBox1_Change( )

b=TextBox3.Text

END

PUBLIC SUB Button1_Click( )

TextBox4.Backcolor=Color.RGB(R, G, B)

END
```

3. 마우스를 이동할 때 그의 위치를 얻는 프로그람을 작성하여라.

<대면부>

<코드>

```
PUBLIC SUb DrawingArea1_MouseMove( )
```

```
TextBox1.Text=Mouse.X

TextBox2.Text=Mouse.Y

END
```

4. 두개의 옹근수를 입력하고 그 값을 각각 x, y자리표로 하는 점을 찍어라.

```
PUBLIC SUB Button1_Click( )

DIM x AS integer

DIM y AS integer

input x

input y

Draw.Begin(DrawingArea1)

Draw.Forecolor=Color.Red

Draw.Point(x, y)

Draw.End

END
```

5. 두개의 점(100, 100), (200, 300)을 맺는 선분을 그리여라.

```
PUBLIC SUB Button1_Click( )

DIM x1, y1, x2, y2 AS integer

x1=100

y1=100
```

```
x2=200

y2=300

Draw.Begin(DrawingArea1)

Draw.Forecolor=Color.Red

Draw.Line(x1, y1, x2, y2)

Draw.End

END
```

6. 점 (100, 100)을 시작점으로 하고 너비가 100, 높이가 200인 직4각형을 그리여라.

<대면부>

<코드>

```
PUBLIC SUB Button1_Click( )
```

```
DIM x, y, w, h AS integer
x=100
y=100
w=100
h=200
Draw.Begin(DrawingArea1)
Draw.Forecolor=Color.Blue
Draw.Rect(x, y ,w, h )
Draw.End
END
```

7. 점 (100, 100)을 시작점으로 하고 너비가 100, 높이가 200인 직4각형구역에 내접하는 타원을 그리여라.

```
PUBLIC SUB Button1_click( )
DIM x, y, w, h AS integer
x=100
y=100
w=100
h=200
Draw.Begin(DrawingArea1)
```

```
Draw.Forecolor=Color.Blue

Draw.Ellipse(x, y, w, h )

Draw.End

END
```

8. 타원을 그리고 그우에 《타원》이라는 문자를 써넣어라.

```
PUBLIC SUB Button1_click( )
DIM x, y, w, h AS integer
x=100
y=100
w=100
h=50
Draw.Begin(DrawingArea1)
Draw.Forecolor=Color.Red
Draw.Rect(x, y, w, h)
Draw.Ellipse(x, y, w, h)
Draw.Text("타원", 150, 150)
Draw.End
END
```

련습문제

1. 원둘레를 그리고 그 아낙을 지정한 형식과 붉은색으로 칠하는 프로그람을 작성하여라.
2. 1~5까지의 수들가운데서 임의의 수를 입력하고 그 값에 해당한 테두리선을 가지는 원둘레를 그리는 프로그람을 작성하여라.

제 5 절. 날아가는 공

이 절에서는 원이 왼쪽에서 오른쪽으로 움직이는 모양을 보여주는 프로그람을 작성하는 과정을 통하여 간단한 동화상의 실현방법에 대하여 인식하도록 한다.

<대면부>

<속성>

조종체이름	속성	값
Button1	Text	공움직이기
PictureBox1	Stretch	True

우선 화상편집기에서 원을 하나 그리고 그것을 파일로 보관한 다음 기본창문의 Data 서류철에 적재한다.

그다음 PictureBox1에 적재하고 다음의 코드를 입력한다.

```
PUBLIC SUB Button1_Click( )
DIM I AS single
FOR I=0 to 300 STEP 0.001
PictureBox1.Move(I, 100)
NEXT
END
```

<코드설명>

PictureBox1.Move(I, 100): PictureBox1조종체를 움직이게 하는 방법이다.

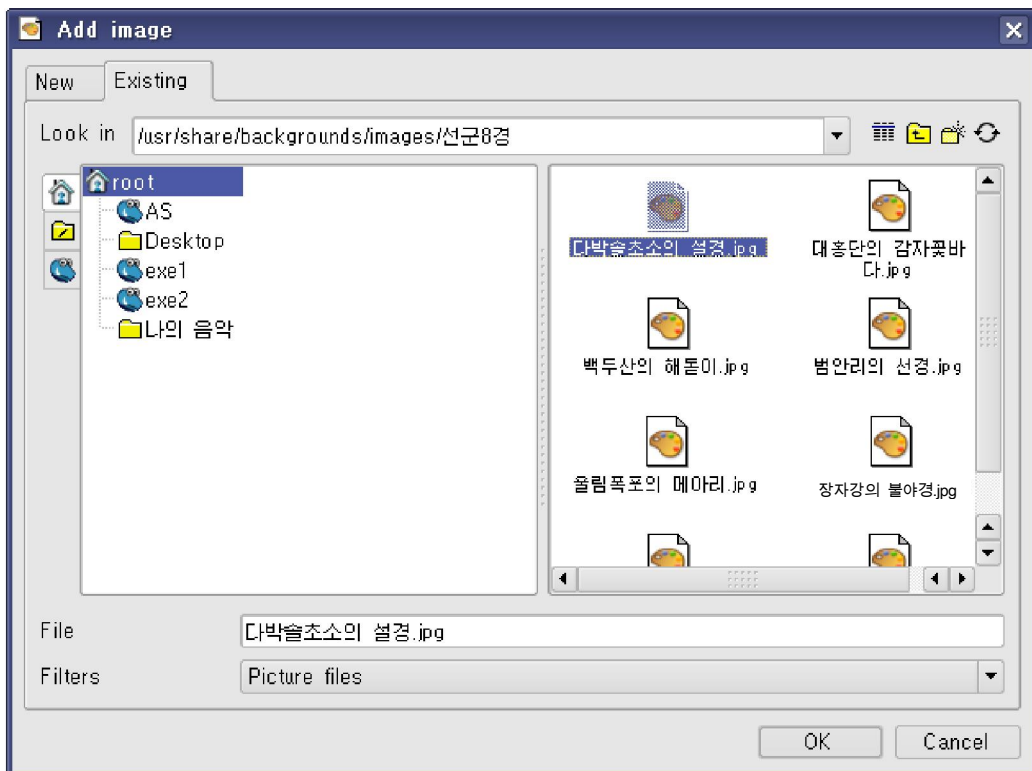
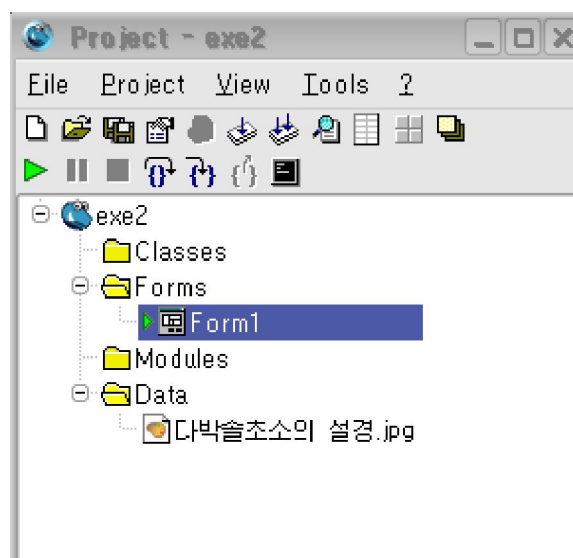
우의 프로그람을 실행시키면 PictureBox1조종체가 X축방향으로 0.001만큼씩 움직인다.

1. 화상조종체

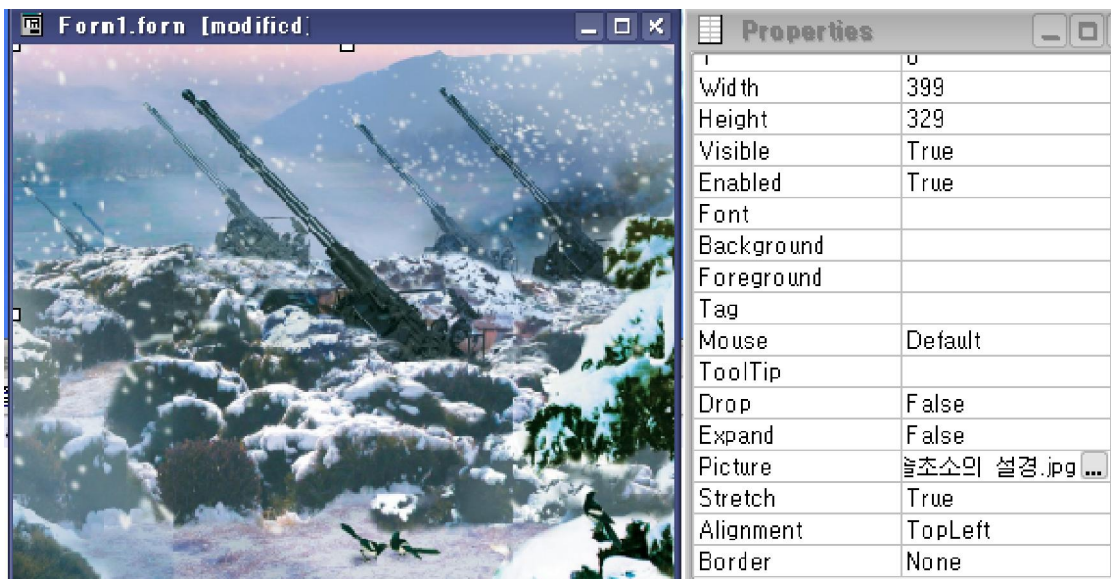
화상조종체(PictureBox) 는 화상파일들을 적재하는 기능을 가진 조종체이다.

실례로 화상조종체에 콤퓨터에 있는 《다박솔초소의 설경》그림을 적재하여보자.

이 조종체에 그림을 적재하기 위하여서는 기본창문의 Data서류철에 그 그림을 미리 적재하여놓아야 한다.

기본창문의 Data서류철을 선택하고 지름차림표를 펼친다. 그다음 New→Image순서로 실행시킨다. 그러면 Add image대화칸이 펼쳐진다.

콤퓨터에 이미 기억되여있는 그림을 추가하려면 Existing표쪽을 찰칵하여 나타나는 대화칸에서 필요한 그림을 선택하고 OK단추를 찰칵한다.

그러면 Data서류철에 그림이 추가되는것을 볼수 있다.

이제 화상조종체에 그림을 적재하여보자.

먼저 화상조종체를 Form설계창문에 배치하고 속성설정칸에서 Stretch속성을 True로 설정한다. 다음 Picture속성칸을 찰칵한다. 이때 나타나는 내리펼침목록을 찰칵하여 나오는 Select a picture대화칸에서 《다박솔초소의 설경》을 선택하고 OK단추를 찰칵하면 화상조종체에 그림이 적재되는것을 볼수 있다.

조종체이동방법의 형식은 다음과 같다.

화상조종체이름.Move(x AS integer, y AS integer [, width AS integer, height AS integ er])

여기에서 x, y는 옮겨간 조종체의 왼쪽웃모서리의 점의 x, y자리표이고 width와height는 옮겨간 다음 조종체의 너비와 높이값들이다.

이 값들을 적당하게 설정하여 멀리 날아가는 공의 모양을 형상할수도 있다.

2. 시간조종체

일정한 간격으로 어떤 동작을 반복시키는 프로그람을 작성할 때 시간조종체(Timer) 를 리용하면 프로그람작성이 간단해진다.

원이 움직이는 프로그람을 이 조종체를 리용하여 작성하여보자.

이 조종체는 Special부분도구칸에 있다.

우선 시간조종체를 Form설계창문에 배치한다.

<속성>

조종체이름	속성	값
Timer1	Enabled	True
	Delay	1 000

<코드>

```
PUBLIC I AS integer
PUBLIC SUB Timer1_Timer( )
I=I+1
```

```
PictureBox1.Move(I, 100)
END
```

우의 프로그람을 실행시키면 마우스로 아무것도 찰칵하지 않아도 자동적으로 사건이 호출되여 실행된다.

다음 Delay속성값으로 준 1 000(1초)동안 있다가 또다시 Timer사건이 호출되여 실행된다. 이렇게 이 사건이 계속 실행된다.

우에서 준 Delay속성값은 2행부터 4행까지를 실행하는데 걸리는 시간이 아니라 2행부터 4행까지를 한번 수행하고 다시 시작하는 시간을 설정한것이라는데 주의하여야 한다.

Enabled속성은 이 조종체가 작용가능하거나 하지 않게 하는 속성이다.

결국 시간조종체는 일정한 시간간격으로 Timer사건을 발생시켜 프로그람을 실행하게 하는 조종체이다.

실 습

1. 멀리 날아가는 공의 모양을 보여주는 프로그람을 작성하여라.

<속성>

조종체이름	속성	값
Button1	Text	날아가는 공
PictureBox1	Stretch	True

화상편집기에서 원을 하나 그리고 그것을 파일로 보관한 다음 기본창문의 Data서류철에 적재한다. 그다음 PictureBox1에 적재하고 다음의 코드를 입력한다.

<코드>

```
PUBLIC SUB Button1_Click( )

DIM I AS single

FOR I=0 to 300 STEP 0.001

PictureBox1.Move(I, 100, PictureBox1.Width/1.5, PictureBox1.Height/1.5)NEXT

END
```

옮겨가는 점의 x자리표를 변화시키면서 동시에 옮겨간 화상조종체의 너비와 높이를 점점 작게 하였다.

이 프로그람을 실행하여보면 옮겨간 다음 너비와 높이가 원래 크기보다 작게 되므로 마치도 멀리 날아가는것과 같이 보이게 된다.

2. 일정한 시간간격으로 본문칸에 콤퓨터안에 있는 시계의 현재시간을 표시하는 프로그람을 작성하여라.

<속성>

조종체이름	속성	값
Timer1	Enabled	True
	Delay	1 000

<코드>

```
PUBLIC SUB Timer1_Timer( )
```

```
TextBox1.Text=Time(Now)

END
```

프로그람을 실행하면 1초간격으로 자동적으로 본문칸에 현재시간이 표시된다.

련습문제

1. 화상조종체에 자기가 그린 그림을 적재하여보아라.
2. 화상조종체 8개를 적재하고 매개에 그림들을 적재하여보아라.
3. 《강성대국》이라는 글발이 왼쪽에서 오른쪽으로 흘러가는 프로그람을 작성하여라.

제 2 장. 3 차원모의프로그람

제 1 절. 3 차원콤퓨터모의기술의 기초

1. 3차원콤퓨터모의의 개념

콤퓨터로 자연계의 물체들을 표현하자면 우선 그것들이 공간물체들이므로 3차원자리표계에서 물체들의 형태를 정확히 표현해야 하며 3차원자리표계에서 표현된 물체들을 콤퓨터의 화면 즉 2차원자리표계에 투영하여야 한다. 뿐만아니라 물체의 밝기와 색 등을 정확히 수자화하여 표시해야 한다.

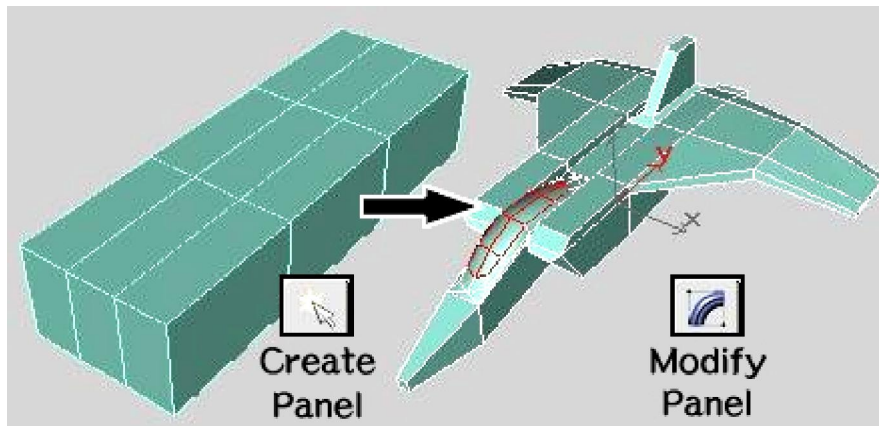
3차원자리표계에서 자연계의 물체를 표현하는것을 모형화(Modeling)라고 하며 그것을 2차원자리표계에 투영하고 물체의 색과 밝기 등을 계산하여 콤퓨터의 화면에 표시하는것을 시각화(Rendering)라고 한다.

콤퓨터로 자연현상을 모형화하기 위해서는 일반적으로 다음과 같이 한다.

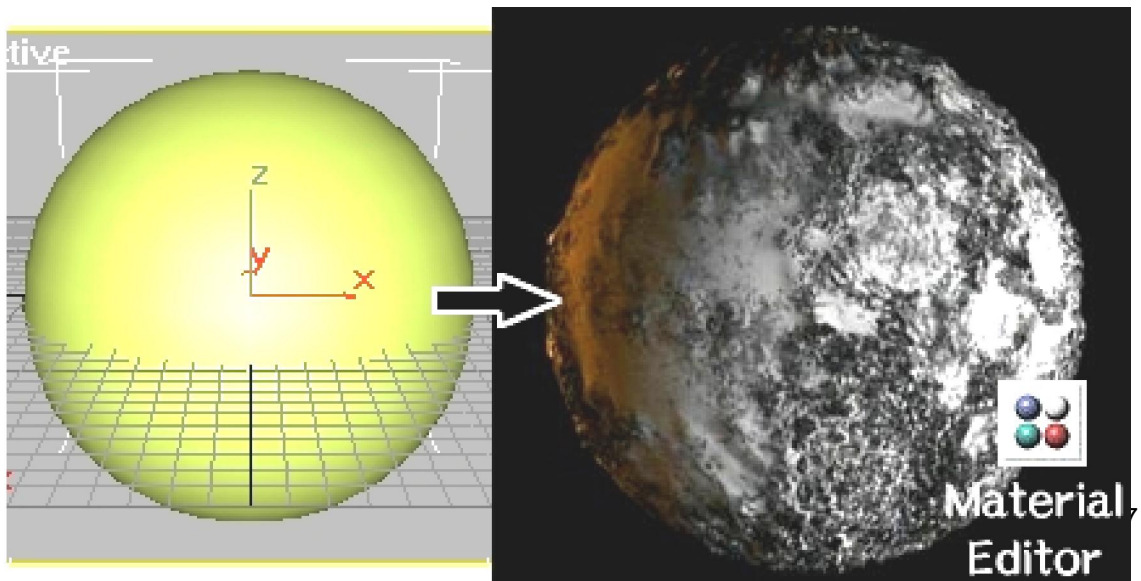
① 모형을 만든다.(Modeling)

자연속의 물체들을 모형화하기 위해서는 먼저 직6면체, 구 등과 같은 기초모형을 만들고 그것이 실지 물체와 비슷해지도록 변형한다.

실례로 3차원화상처리프로그람인 3ds max로 비행기화상을 만들 때 창작판에서 기초모

형으로 직6면체를 만들고 그것을 변형판에서 비행기형태로 변형시킨다.

② 모형에 재질을 입힌다.(Mapping)

물체를 모형화한 다음에는 겉면에 재질도 주어야 보다 실감있게 보인다.

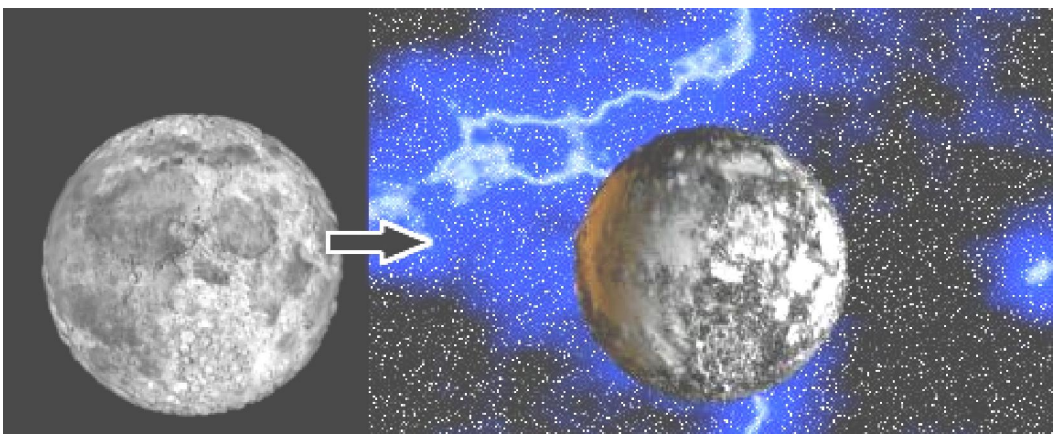
실례로 3ds max에서 달화상을 만들 때 달의 기초모형으로 구를 만들고 여기에 재질편성기에서 달겉면의 굴곡을 표현하는 재질을 입힌다.

③ 모형에 움직임을 준다.(Animation)

모형을 만들고 모형에 움직임을 주어야 한다.

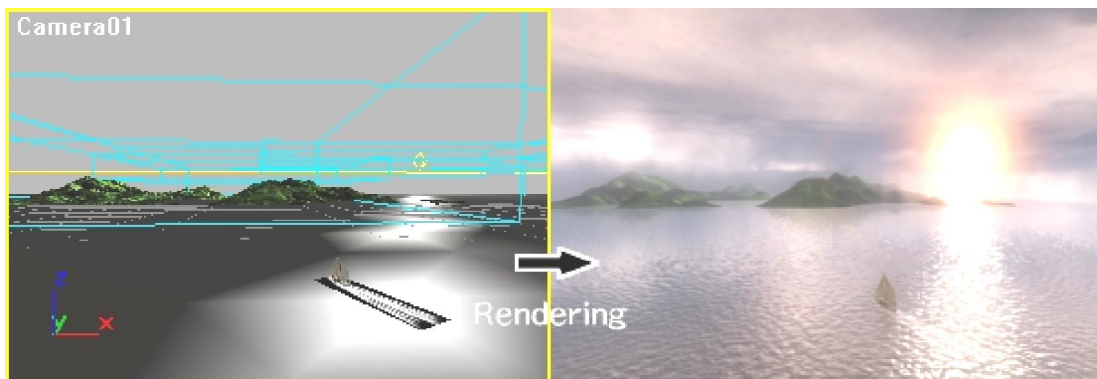
실례로 3ds max에서 달화상을 만든 다음 그것의 움직임은 자리길보기(Track View)에서 준다. 여기서는 자리표값의 변화뿐아니라 모형의 크기, 모양, 색 등을 변화시킨다.

④ 모형의 주위환경을 만든다.(Environment)

움직임을 주고 재질을 준 다음에는 모형이 어떤 주위환경에 있는지를 묘사해주어야 한다.

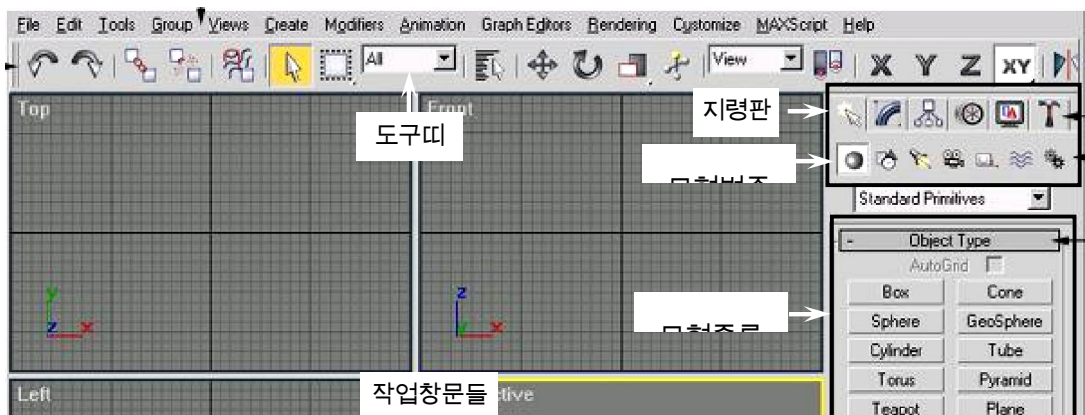
실례로 3ds max에서 이것을 환경묘사(Environment)에서 한다.

⑤ 장면을 파일로 출력한다.(Rendering)

작업창문에서 만든 모형들을 시각화하여야 화상 혹은 동화상파일로 만들수 있다.

3ds max에서는 Rendering\Render지령을 리용하여 장면을 시각화한다.

콤퓨터에 의한 화상처리

콤퓨터에 의한 최초의 도형표시체계는 1963년에 개발된 대화형 2차원도형표시체계인 Sketchpad이다. 당시에는 현시장치에서 도형을 립체적으로 표시하지 못하였기때문에 자동차나 배, 비행기 등을 설계할 때 2차원제도에서 리용되였다. 1970년대에 들어와 지금과 같은 현시장치가 개발되여 모형의 색과 농담 등을 리용하여 그것을 립체적으로 표시할수 있게 되였다. 이와 같이 콤퓨터로 화상을 생

2. 3ds max의 화면구성

3ds max를 기동하면 다음과 같이 4개의 구역으로 갈라진 창문이 펼쳐진다.

3ds max에서 작업파일의 확장자는 max이다. 특징적인 차림표들은 다음과 같다.

Group - 모형들을 묶어 관리하기 위한 지령들이 들어있다.

Views - 작업창문의 보기와 관련한 지령들이 들어있다.

Create - 모형을 창조하기 위한 도구들이 들어있다.

Modifiers - 창조된 모형에 대한 변형도구들이 들어있다.

Character - 골조를 만들기 위한 도구들이 있다.

Animation - 움직임을 설정하기 위한 지령들이 들어있다.

Graph Editors - 움직임을 그라프로 주기 위한 지령들이 들어있다.

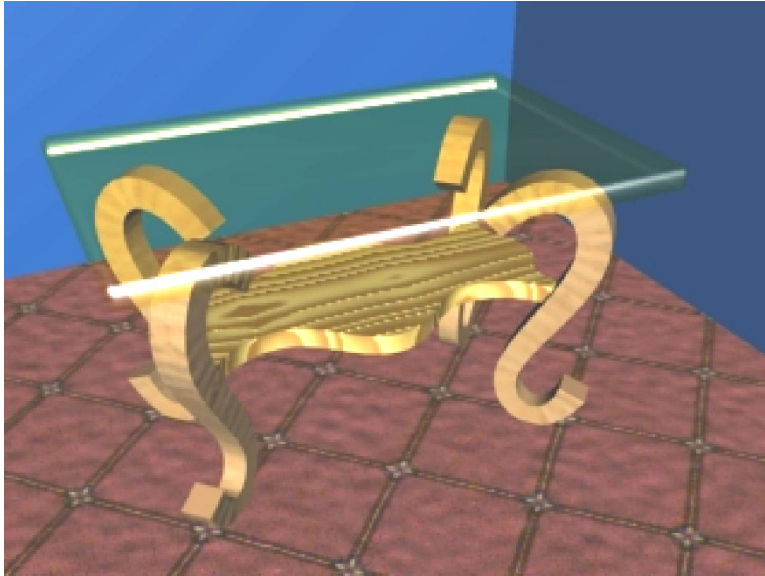
Rendering – 창조된 모형을 동화상파일로 작성하는 도구들과 환경설정, 재질설정 도구들이 있다.

Customize - max의 환경을 설정하기 위한 지령들이 들어있다.

MaxScript - max의 자체언어를 리용하여 동화상을 창작하기 위한 지령들이 들어있다.

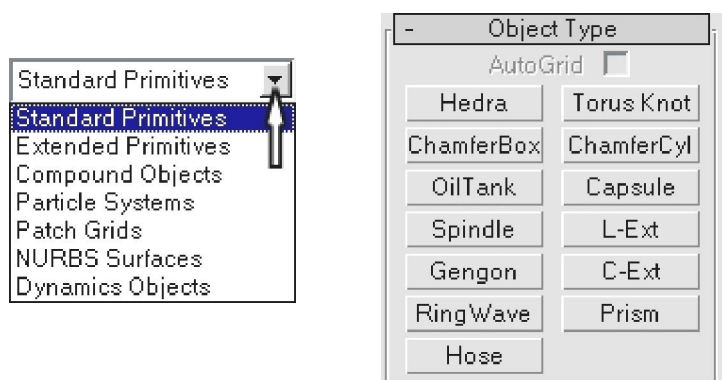
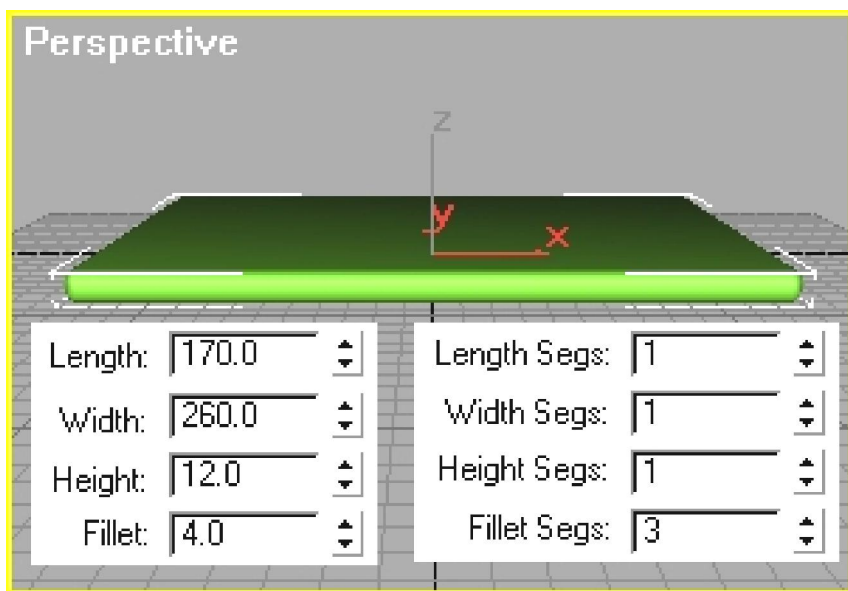
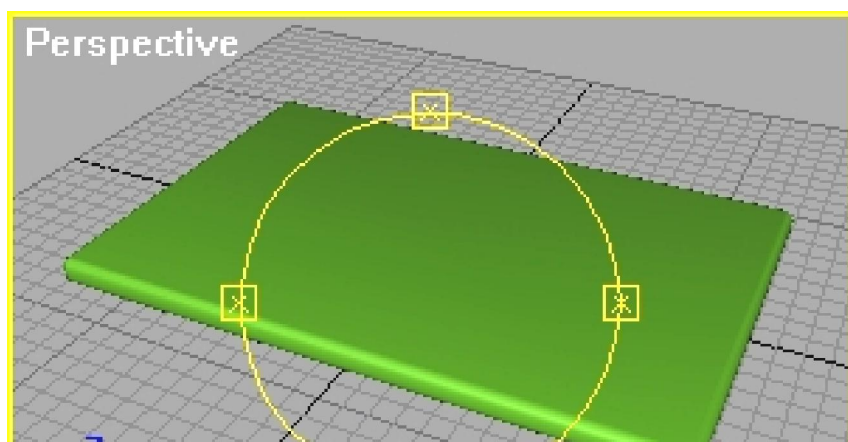
3ds max의 새로운 기능들가운데서 가장 중요한것은 지령판이다. 지령판에는 편집과 그림그리기에 관한 모든 지령이 보관되여있다.

제 2 절. 탁자만들기

여기서는 다음 그림과 같은 모양의 탁자를 만드는 과정을 통하여 모형화작업과 모형에 재질을 입히는 방법을 학습하도록 한다.

1. 모형화작업

1) 먼저 탁자웃판부분을 만든다.

기초모형을 만들기 위하여 지령판(Command Panel)의 창작판(Create Panel) 을 찰칵하고 그아래에 나타나는 모형범주에서 Geometry(기하학적도형) 를 선택한다.

그리고 아래와 같이 Standard Primitives(표준기초모형)옆에 있는 검은 3각형을 찰칵하여 나타나는 목록에서 Extended Primitives(확장된 기초모형)를 선택하면 여기서 만들수 있는 모형들의 종류가 모형종류(Object Type)에 나타난다.

여기서 Chamfer Box(모죽임6면체)를 선택하고 마우스지시자를 작업창문에 가져가서 아무런 한 점을 찰칵하고 끌기하여 자기가 바라는 형태를 만들고 다시 찰칵하는 방법으로 모죽임6면체를 만든다. 이때 Parameters(파라메터)에서 아래그림과 같이 값들을 지정해준다.

Perspective(투시)창문에서 만들어진 모형을 살펴본다.

작업창문조종판 에서 (Zoom: 확대)나 (Arc Rotate: 돌려보기)를 리용하여 모형을 살펴본다.

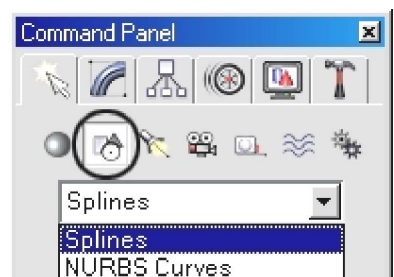
Fillet는 모서리를 얼마나 둥글게 처리하는가를 지정한다.

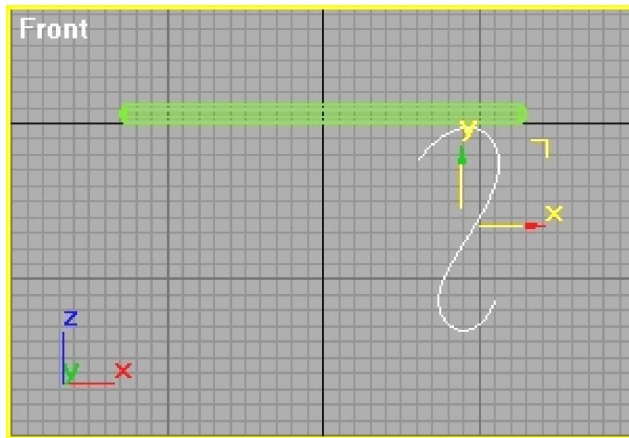
Fillet Segs는 모서리의 토막수를 지정한다.

둥근 부분을 직선토막들의 결합으로 만드는데 이때 리용되는 직선토막들의 수가 Segs이다. 그러므로 Fillet Segs값이 크면 보다 원활해지고 값이 작으면 각지게 된다.

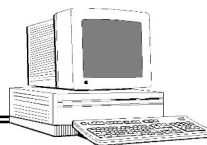
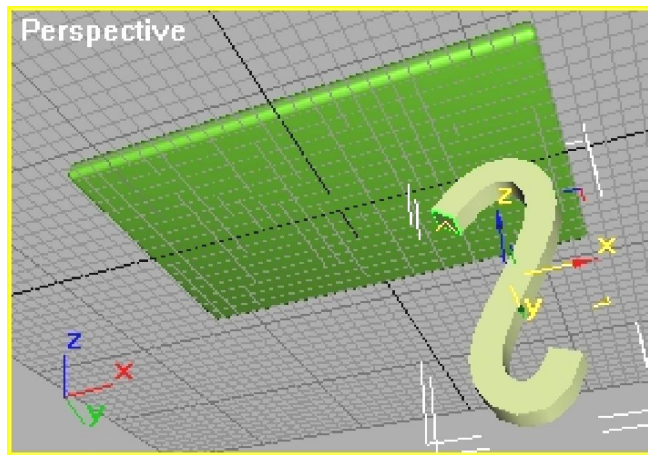
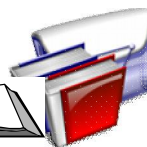
2) 다음 탁자다리를 만든다.

먼저 창작판의 모형범주에서 Shapes(모양) 를 리용하여 선을 그린다.

Shapes를 선택하면 그아래의 목록에는 Splines (스플라인곡선)와 NURBS Curves(NURBS곡선)항목이 있다.

Splines를 선택하면 모형종류에 곡선우에 있는 점들과 방향벡토르에 의해 모양이 결정되는 11개 종류의 스플라인곡선항목들이 나타난다.

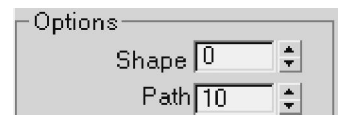
여기서 Line(선)을 선택하고 작업창문에서 마우스를 끌기하여 탁자다리모양을 그린다.

이것은 NURBS곡선으로도 그릴수 있다. 다음 탁자다리의 자름면과 같은 4각형을 하나 그린다. 그리고 먼저 그린 곡선을 선택한 다음 창작판의 모형범주에서 Geometry(기하학적 도형) 를 선택하고 그아래의 목록에서 Compound Objects(혼합모형)를 찰칵한다. 그리

고 모형종류에서 Loft를 선택한다. Loft는 어떤 경로를 따라 선택한 선을 이동시켜 모형을 만드는 기능이다.

다음 그아래에 펼쳐지는 차림표에서 Get Shape(모양주기)를 찰칵하고 탁자다리의 자름면으로 그린 4각형을 선택하면 다리형태가 만들어진다.

Splines에 있는 곡선들을 생성하는 방법은 Geometry에서와 비슷하다. 그러나 이것들은 형태만 가지고있을뿐 두께가 없다. 그러므로 시각화하여도 보이지 않는다.

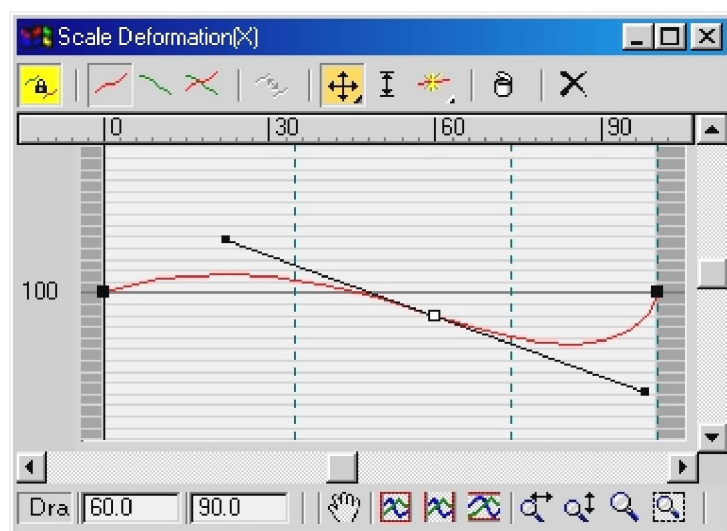
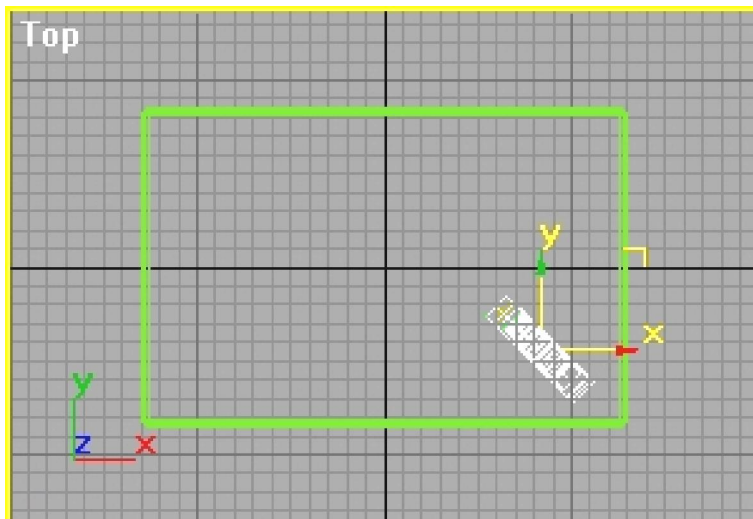
모형종류에서 만들려는 선의 종류를 선택하면 Rendering이라는 항목이 나타난다. 여기서 Renderable을 선택해주면 Thickness(두께)에 값을 준 두께로 시각화된다.

만들어진 모형의 변형은 지령판의 변형판(Modify Panel) 에서 진행한다.

탁자다리모형을 완성하기 위하여 변형판의 Skin Parameters항목에서 Shape(모양)와 Path(경로)의 설정 값을 0, 10으로 변경시킨다.

그리고 DeFormations(기울이기)항목에서 Scale(척도)을 선택한다.

그러면 Scale DeFormation(X)대화칸이 열린다.

여기서 Loft할 모형의 굵기를 그라프로 조절할수 있다.

조종점을 삽입하고 다음 그림과 같이 조절한다.

기본도구띠에서 이동도구 (Move)와 회전도구 (Rotate)를 리용하여 탁자다리를 해당한 위치에 옮겨놓는다.

기본도구띠의 Mirror(거울)도구 를 리용하여 탁자다리를 대칭복사한다.

탁자다리를 선택하고 를 찰칵하고있으면 이 나타나는데 여기서 를 선택한다.(이것들은 탁자다리를 대칭복사할 때 중심축을 어디에 두겠는가를 설정하기 위한것이다.) 다음 Mirror Axis(거울축)에서 X, Clone Selection(단순복사선택)에서 Copy로 선택한다. 다음 OK단추를 찰칵한다. 그러면 탁자다리가 대칭복사된다. 같은 방법으로 탁자다리 4개를 만든다.

이렇게 하여 탁자다리를 다 만들었다.

기초모형을 만든 다음에는 그것을 실물과 일치하도록 변형시켜야 한다.

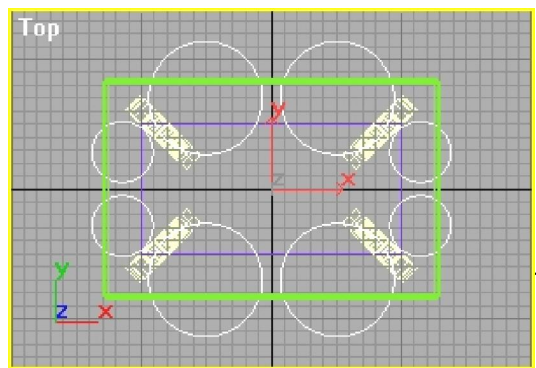
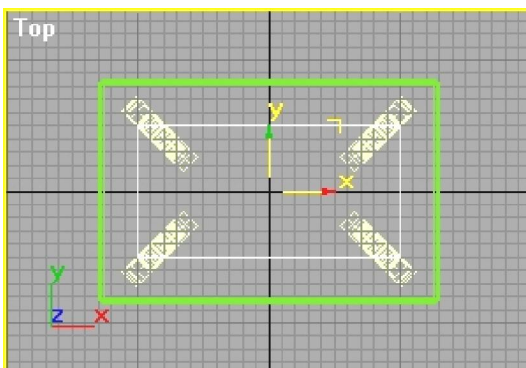
례를 들어 사과를 만들려고 하는 경우 구를 만들어놓고 그것을 사과모양으로 변형시켜야 한다. 이런 변형작업을 지령판의 변형판에서 한다.

변형체목록(Modifier List)을 찰칵하면 변형체들이 표시된다.

여기서 적용하려는 변형체를 선택하면 그 변형체의 설정항목이 나타난다. 추가된

3) 이제는 탁자밑받치개를 만든다.

먼저 Shapes의 Splines를 선택하고 모형종류에서 Rectangle(직4각형)을 찰칵한다. 그리고 작업창문에서 탁자의 웃면보다 작은 직4각형을 하나 그린다.(아래 왼쪽그림)

다음 Circle(원)을 찰칵하고 아래의 오른쪽그림에서와 같이 원을 그린다.

직4각형을 선택하고 변형판에 가서 변형체목록에서 Edit Spline(스플라인편성)을 찰칵한다.

Edit Spline의 Geometry항목에서 Attach(부속)를 선택한다.

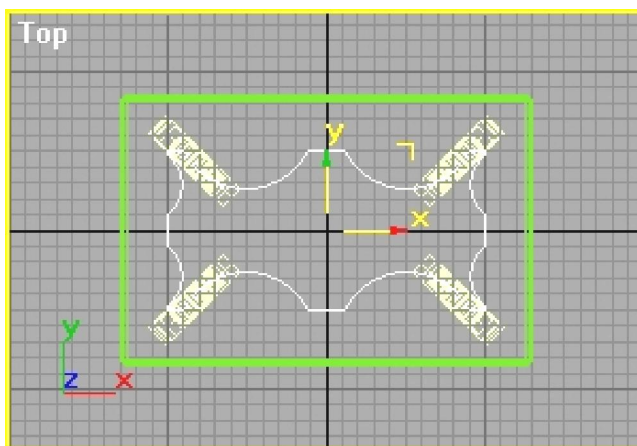
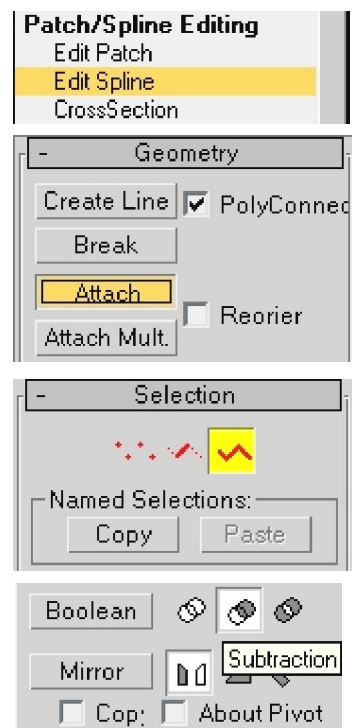
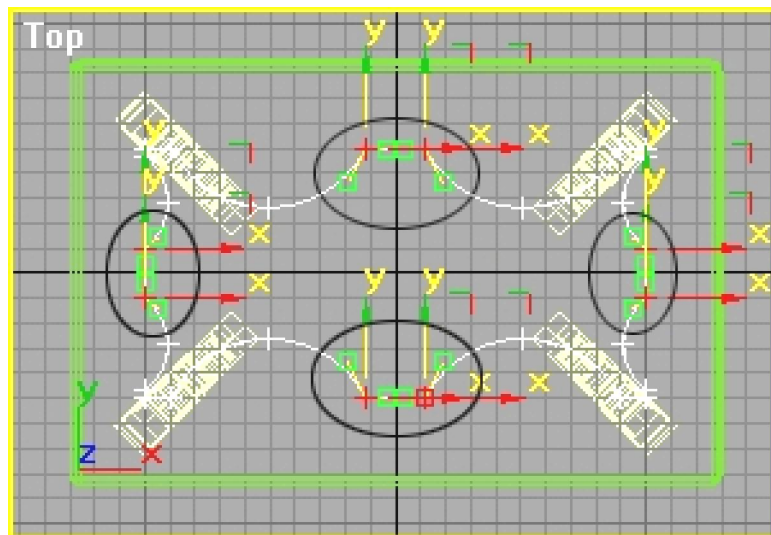
Attach는 여러개의 Shape를 붙여서 하나의 Shape로 만드는 기능을 수행한다. 8개의 원을 직4각형에 붙인다.

Attach가 선택된 상태에서 마우스지시자를 원에 가져가면 모양이 달라지는데 이때 찰칵하면 된다.

Selection항목에서 Spline을 선택한다.

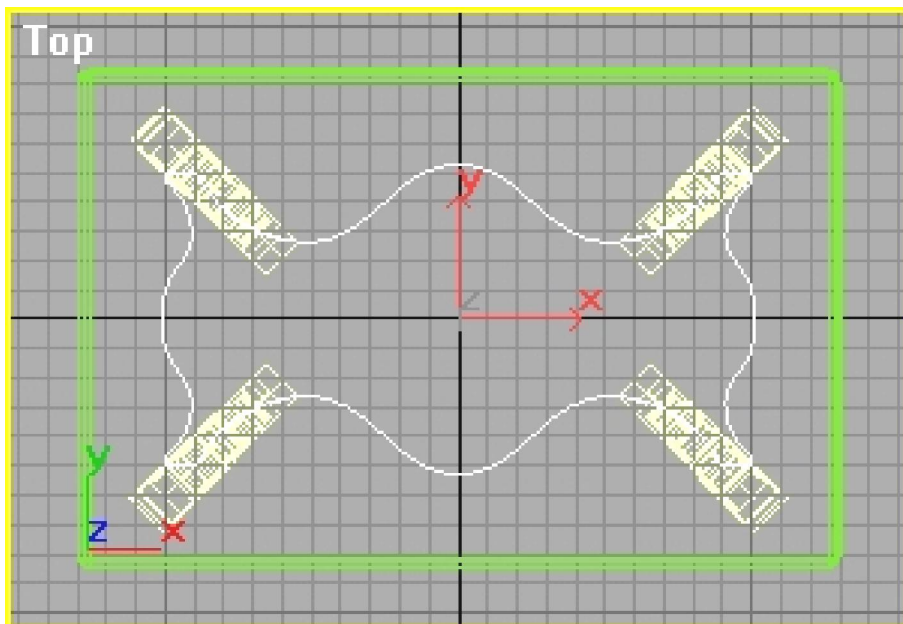
직4각형을 선택한다. 그러면 직4각형이 붉은색으로 된다.

Geometry항목에서 Boolean(모임연산)을 선택하고 Subtraction(삭제)을 찰칵한다.

8개 원을 하나씩 차례로 선택해준다. 그러면 원이 없어지면서 겹치는 부분이 잘리워진다.

변가운데 있는 8개 정점을 선택한다.

마우스지시자를 선택된 정점들가운데 어느 한 정점우에 가져가서 오른쪽찰칵하면 다음과 같은 차림표가 나온다.

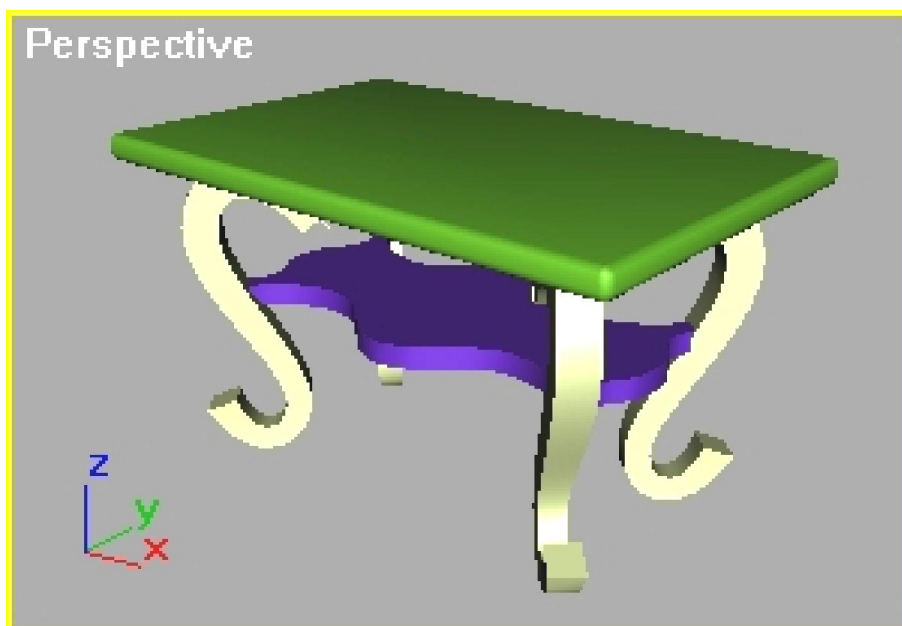
Reverse Spline
Make First
Divide
Bind
Unbind
Bezier Corner ✔
Bezier
Corner
Smooth
Sub-objects
TOOLS 1
TOOLS 2
Create Line
Attach
Detach Segment
Connect
Refine
Refine Connect
Cycle Vertices
Break Vertices
Weld Vertices

Unfreeze All
Freeze Selection
Unhide All
Hide Unselected
Hide Selection
Unhide All (Spline)
Hide (Spline)
DISPLAY
TRANSFORM
Move
Rotate
Scale
Manipulate
Properties...
Track View Selected
Wire Parameters
Convert To:

여기서 Smooth(다듬기)를 선택한다. 그러면 경계가 유연해진다.

변형판의 변형체목록에서 Mesh Editing(그물편성)\Extrude(돌출)를 실행한다.

Mesh Editing
DeleteMesh
Edit Mesh
Extrude
Face Extrude

Amount(층수)를 10으로 준다. 탁자밑받치개의 위치를 조절해준다.

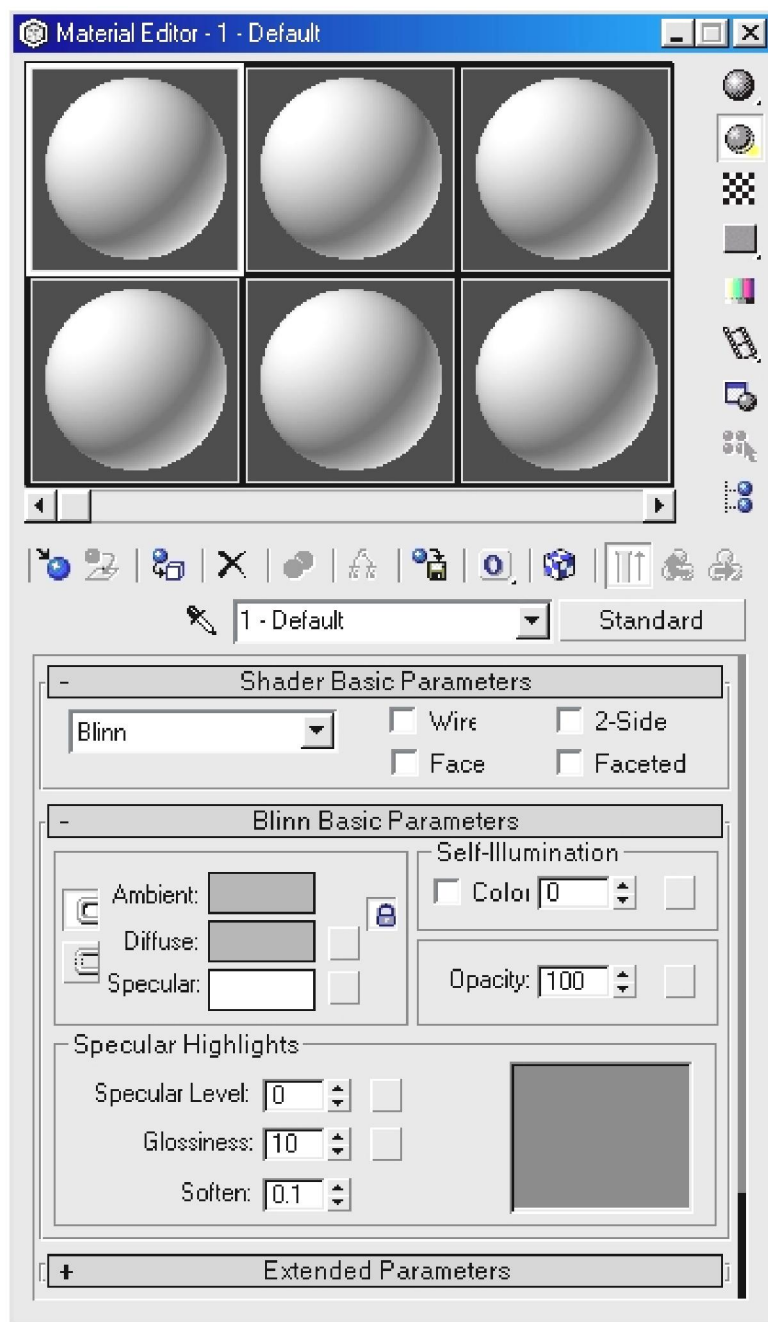
결과는 다음과 같다.

이렇게 하여 탁자모형을 만들었다.

2. 재질입히기

재질(Material)은 모형들이 현실처럼 보이게 하는데서 아주 중요하다.

재질입히기(Mapping)는 재질을 모형의 겉면에 투영하는 방법이다.

재질과 관련된 작업은 재질편성기(Material Editor)에서 한다.

기본도구띠의 를 찰칵하거나 건반에서 M건을 누르면 재질편성기가 열린다.

재질편성기를 열면 제일 첫번째 구(Sample Slot: 견본)에 흰 경계선이 있는것을 볼수 있다. 이것은 그것이 선택되여있다는것이다.

앞에서 만든 탁자에 재질을 입혀보도록 한다.

1) 먼저 탁자다리에 나무재질을 입힌다.

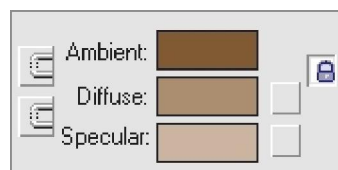
재질편성기의 도구띠에서 (Get Material: 재질입히기)를 찰칵하여 Material\Map Browser(재질/사영체열람기)대화칸을 연다.

Browse From에서 Mtl Library(재질서고)를 선택한다. Wood_Ash(Standard)를 두번찰칵한다. 이것은 표준물푸레나무목재의 재질이다.

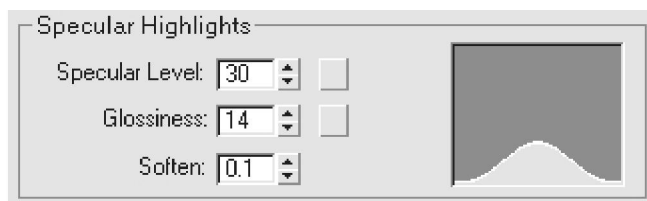
탁자다리를 모두 선택한다. 재질편성기에서 (Assign Material To Selection: 선택부분에 재질할당)을 찰칵하여 탁자다리에 재질을 입힌다.

2) 다음 탁자 밑받치개에 나무재질을 입힌다.

재질편성기에서 두번째 구를 찰칵한다.

Diffuse(발산)옆의 4각형을 찰칵하여 Color Selector (색선택기)를 연다. Diffuse Color(발산하는 색)를 적당한 나무색으로 설정해준다.

Ambient Color(주위의 색)를 더 어두운 색으로 설정해준다.

Specular Color(반사되는 색)를 Diffuse Color보다 밝은 색으로 설정해준다.

Specular Highlights(반사되는 기본점)를 다음과 같이 설정해준다.

Maps항목을 펼치고 Diffuse Color줄의 None을 찰칵한다.

Material\Map Browser에서 Wood(목재)를 선택하고 OK단추를 찰칵한다.

(Go to Parent: 본래로 가기)를 찰칵하고 Diffuse Color줄의 Map#1(Wood)를 Bump(부딪침)줄의 None으로 끌고간다. 탁자의 밑받치개를 선택하고 을 찰칵하여 재질을 입힌다.

재질편성기의 아래단에는 파라메터목록들이 있다.

Blinn Basic Parameters목록을 보면 3개의 부분으로 되여있다.

첫 부분에서는 재질의 색을 설정한다.

Ambient Color: 빛을 받지 못하는 부분의 색

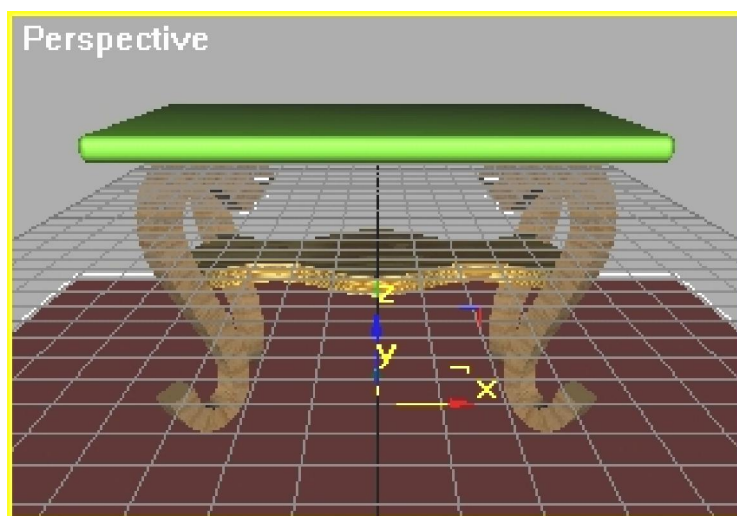
Diffuse Color: 빛을 받는 부분의 색 즉 물체자체의 색

Specular Color: 빛을 받아 반사하는 부분의 색

둘째 부분에서는 물체가 내는 빛과 투명도를 설정한다.

Self-Illumination(자체조명)에서는 물체가 스스로 빛을 내는 효과를 만들어준다. Color를 선택하면 그 색으로 만들어주고 Color를 선택하지 않으면 Diffuse색으로

3) 다음 바닥을 만들고 재질을 입힌다.

창작판의 Geometry에서 Box(직6면체)를 선택하고 탁자가 놓이는 바닥을 만든다.

재질편성기에서 세번째 구를 선택한다.

Maps(사영체)항목을 펼치고 Bump줄의 None을 찰칵한다.

Material\Map Browser에서 Noise를 선택하고 OK단추를 찰칵한다. 이때 시각화해보면 알수 있는바와 같이 바닥이 지나치게 울퉁불퉁하다.

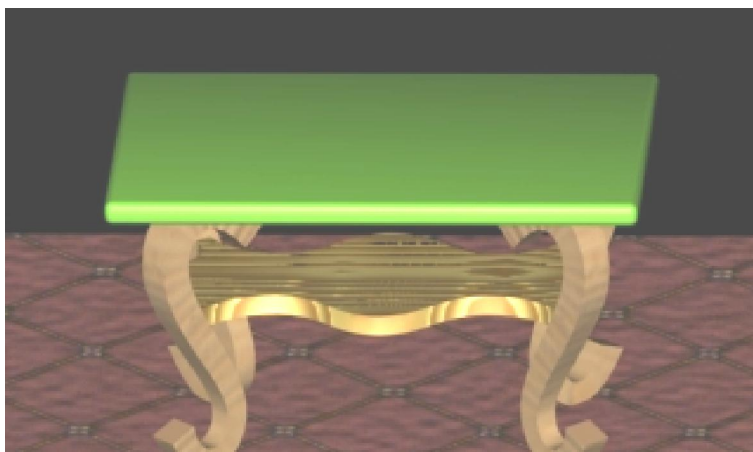
Noise Parameters에서 Size를 10으로 설정한다.

를 찰칵하고 Maps항목에서 Diffuse Color를 찰칵한다.

Material\Map Browser에서 Bitmap를 선택하고 OK단추를 찰칵한다.

3ds max 4\Maps\Fabric\Pat0039.tga파일을 선택한다.

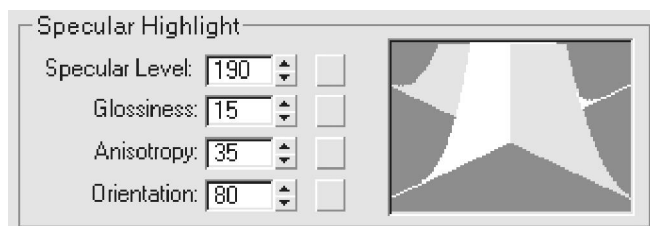
Coordinates의 UV Tiling에 각각 5를 입력한다.

시각화한 결과는 다음과 같다.

4) 다음 탁자웃면에 유리재질을 입힌다.

재질편성기에서 네번째 구를 선택한다.

Shader Basic Parameters에서 Anisotropic(방향 같지 않음)를 선택한다.

Diffuse와 Ambient는 검은색, Specular는 흰색으로 설정한다. 유리를 표현할 때 대체로 Diffuse와 Ambient는 검은색, Specular는 흰색으로 설정한다.

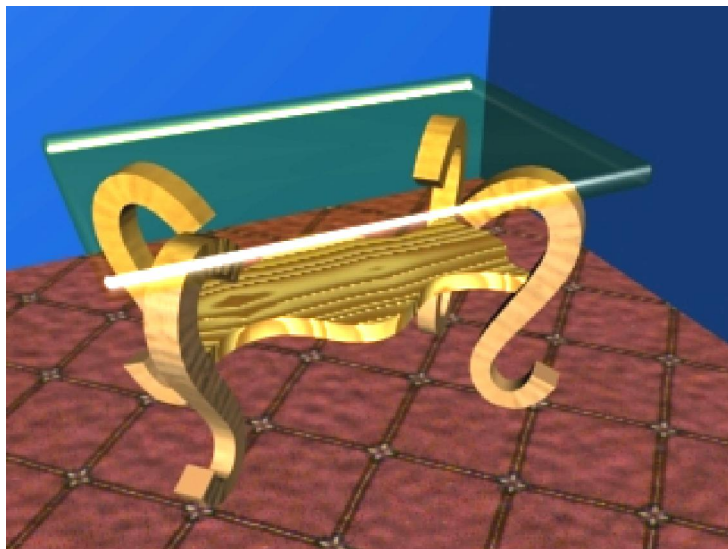
Specular Highlight를 다음과 같이 설정한다.

Opacity를 40으로 설정한다. Extended Parameters에서 Fall off를 In으로, Amt를 50으로 설정한다. Filter(려과기)에서 유리의 색을 설정해준다.

Shader Basic Parameters에서 2-Side를 선택한다.

시각화한 결과는 다음과 같다.(여기서 벽체를 만들어 세워주었다.)

이렇게 하면 탁자가 완성된다.

알아두기

Blinn
Anisotropic
Blinn
Metal
Multi-Layer
Oren-Nayar-Blinn
Phong
Strauss

재질편성기의 아래단에는 파라메터목록들이 있다.

재질편성기의 Shader Basic Parameters에서 Blinn의 옆에 있는 검은 3각형을 찰칵하면 항목들이 나타난다. 이것들은 Shader종류들이다.

Shader는 겉면시각화를 어떻게 계산하겠는가를 알려주는 알고리듬이다.

Shader는 자기의 기능을 따서 이름지어지기도 하고 또는 람

Metal: 금속모형을 만들기 위하여 리용한다.

Strauss: 금속의 금속성정도를 조절할수 있게 한다.

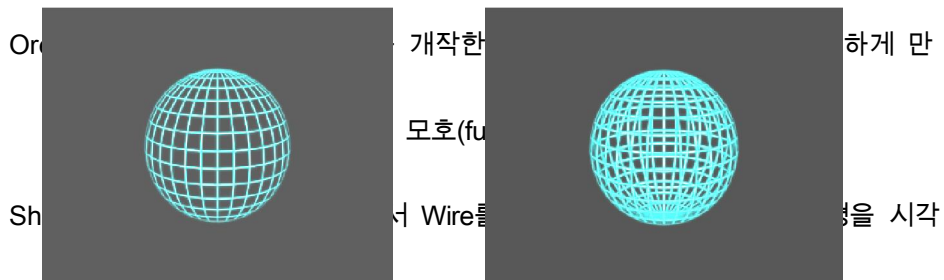
Phong: 면들사이를 고르롭게 처리할수 있게 한다. 질감, 볼록감, 투명도 등과

같은 재질입히기형태들을 취급한다.

Blinn: Phong Shader와 같은 속성을 가지지만 보다 정확하다. 표준Shader이다.

Or 개작한 하게 만

모호(fu

Sh 서 Wire를 형을 시각

화한다. 그리고 2-Side를 선택하면 모형의 안쪽면도 나타난다.

이처럼 지 ... 다.

재질편성 ... 같이 된다.

Self-Illumination: Self-Illumination설정값에 Map를 적용한다.

Opacity: Opacity의 세기를 Map로 결정한다. 검은색에 가까울수록 낮은 값이 적용되어 모형이 투명해진다.

Filter Color: 투명한 모형에 적용되어 투명재질의 색을 표현해준다.

Bump: 모형을 울퉁불퉁한것처럼 보이게 한다

련습문제

1. 표준기초모형(Standard Primitives)에 있는 모형들을 모두 만들어보아라.
2. 책상우에 공이 놓여있는 장면을 만들어라.
3. 확장된 기초모형(Extended Primitives)에 있는 모형들을 모두 만들어보아라.
4. Gengon을 리용하여 5각기둥, 6각기둥을 만들어보아라.
5. Line(선)을 리용하여 조선지도를 만들어보아라.
6. Star(별)를 리용하여 5각별을 만들어보아라.
7. Star의 Distortion(표시령역변형)을 변화시켜 전기톱날을 만들어보아라.
8. 구가 용수철에 매달려 우아래로 흔들리는 장면을 만들어보아라.
9. NURBS Curve를 리용하여 꽃잎을 만들어보아라.
10. Connect(련결)를 리용하여 탁구채를 만들어보아라.
11. Shape Merge(형태병합)를 리용하여 직6면체의 웃면에 자기 이름으로 된 구멍을 만들어보아라.
12. Terrain(지형)을 리용하여 여러가지 형태의 산을 만들어보아라.

제 3 절. 달모형만들기

여기서는 구체적인 표면과 주위환경을 묘사한 달의 운동을 만들어보도록 한다.

1. 달만들기

1) 먼저 달의 기초모형을 만든다.

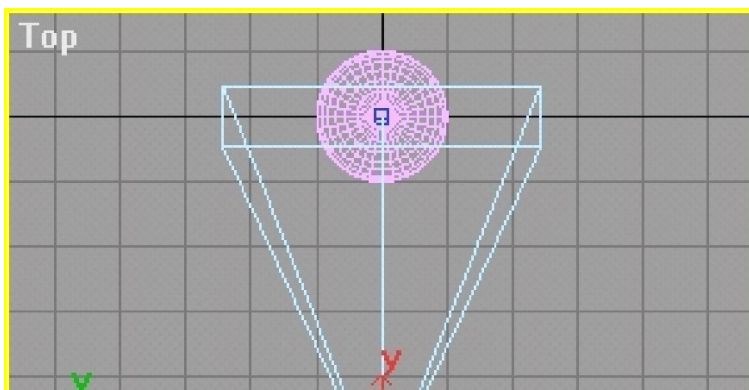
① Perspective(투시)창문에서 구를 하나 만든다.

창작판의 모형범주에서 를 찰칵하고 Standard Primitives를 선택하면 표준기초모형목록이 나타난다. 여기서 Sphere(구)를 선택하고 Perspective창문에서 마우스를 끌기하여 구를 만들수 있다. 이때 구의 이름을 Earth's Moon(지구의 달), Radius(반경) = 100으로 설정한다.

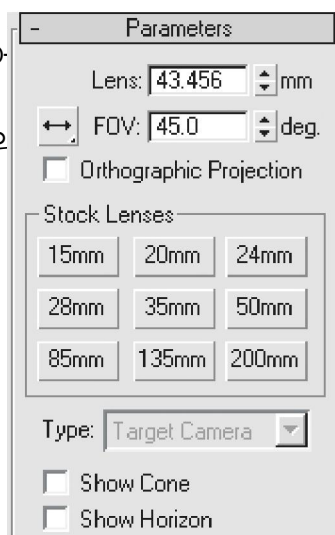
② 작업창문조종판 에서 (Zoom Extents All)을 찰칵한다.

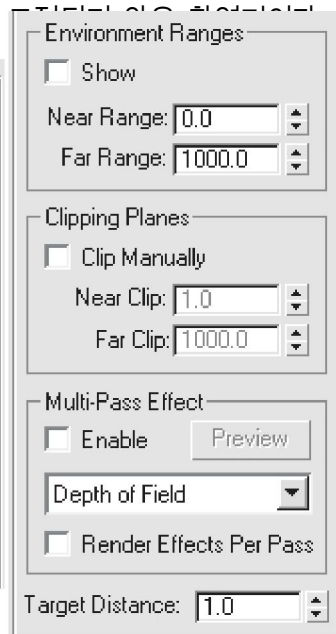
③ 창작판의 모형범주에서 (Cameras: 촬영기)를 찰칵하고 아래에 펼쳐지는 모형종류(Object Type)에서 Target(표적)를 선택한다.

④ Perspective창문에서 임의의 곳을 끌기하여 Target Camera를 만든다.
⑤ 건반에서 Ctrl+C건을 눌러 촬영기를 Perspective창문에 맞춘다. 다음 C건을 눌러 Perspective창문을 Camera창문으로 바꾼다.
⑥ 작업창문조종판에서 Zoom Extents All을 찰칵한다.

Top창문에서 보면 카메라가 Y축방향에서 달을 향하여 놓여있다.

알아두기

Camera에는 2가지 종류가 있다. Target Camera(표적촬영기)는 끝점이 고정된 촬영

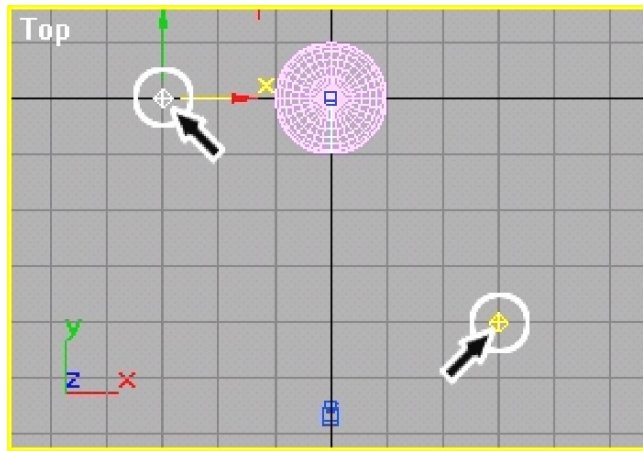
기이고 Free Ca

작업창문 Camera창문으로

된다. Camera

실지 촬영기로 대상을 찍을 때 초점을 맞춘 부분은 뚜렷하게 나오고 나머지 부분에서는 초점이 흐려지는것과 움직이는 모형을 찍을 때 모형이 선명하게 찍혀지지 않는 현상을 Depth of Field(마당깊이)와 Motion Blur(움직임흐리기)를 리용하여 표현한다. Multi-Pa e(권한 각화할 때 효과

⑦ 창작판의 모형범주에서 (Lights: 조명)을 찰칵하고 아래에 펼쳐지는 모형종류 (Object Type)에서 Omni(초점없이)를 선택한다.

⑧ Top(웃부분)창문의 오른쪽아래구석을 찰칵하여 Omni조명을 만들고 이름을 main light(기본조명)로 바꾼다.

⑨ 변형판의 Generation Parameters목록에서 Cast Shadow를 선택하고 Multiplier를 1.2로 설정한다.

⑩ Top창문에서 달의 왼쪽을 찰칵하여 또 다른 조명을 만들고 이름을 fill light(보조조명)로 바꾼다.

조명(Light)은 태양빛, 전등빛 등의 효과를 만들어내는 모형이다. 서로 다른 종류의 조명들은 서로 다른 효과를 만들어낸다. 만일 화상장면(Scene)에 조명이 없으면 3ds max는 Default(이미 정해져있는)조명을 만들어 리용한다. 화상장면을 보다 실감있게 만들기 위해서는 조명을 직접 설치해주어야 한다. 만일 조명을 따로 만들면 Default조명은 없어진다.

조명종류에는 Omni, Spot(집중), Direction(지향성)의 3가지가 있다.

Omni는 백열등과 같이 모든 방향으로 뻗어나가는 조명을 만든다.

Target Spot(표적집중)는 지정된 점을 향하여

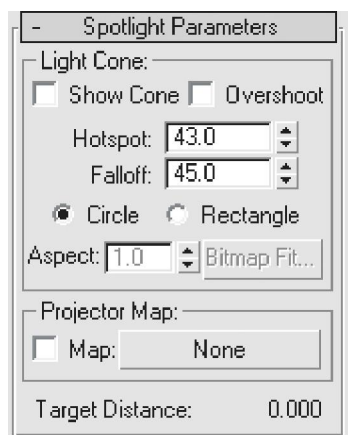
General Parameters

조명은 모형의 빛을 받는 부분(Diffuse)과 정반사하는 부분(Specular)에 모두 영향을 미친다. 그러나 Diffuse나 Specular의 선택을 없애면 그 부분에는 영향을 미치지 않는다. Ambient Only를 선택하면 Ambient부분에만 영향을 미친다.

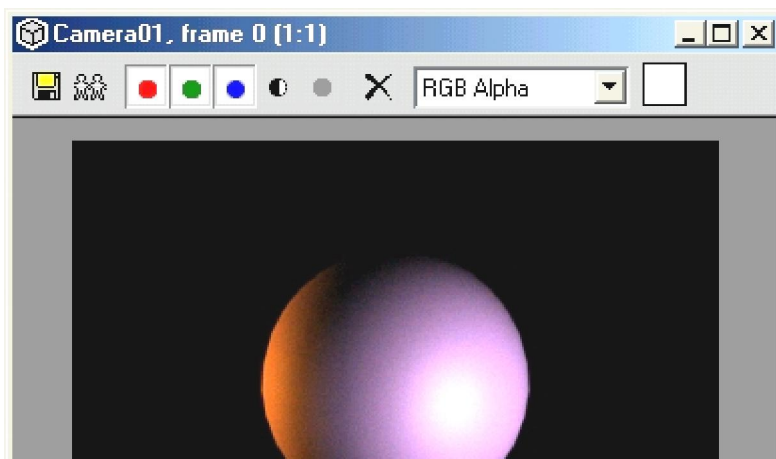
② Spotlight Parameters

Spot에만 있는 항목이다.

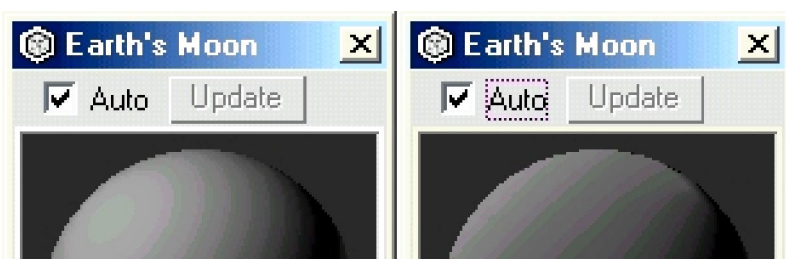
Show Cone을 선택하면 빛줄기의 폭과 길이를 보여준다. Overshoot를 선택하면 Omni처럼 빛이 사방으로 퍼져나간다.

Hotspot에서는 빛줄기의 가장 밝은 폭의 크기를 설정한다. Hotspot바깥쪽으로 가면서 빛이 희미

⑪ 변형판의 General Parameters목록에서 Cast Shadow를 선택하고 Multiplier를 1.5로 설정한다.

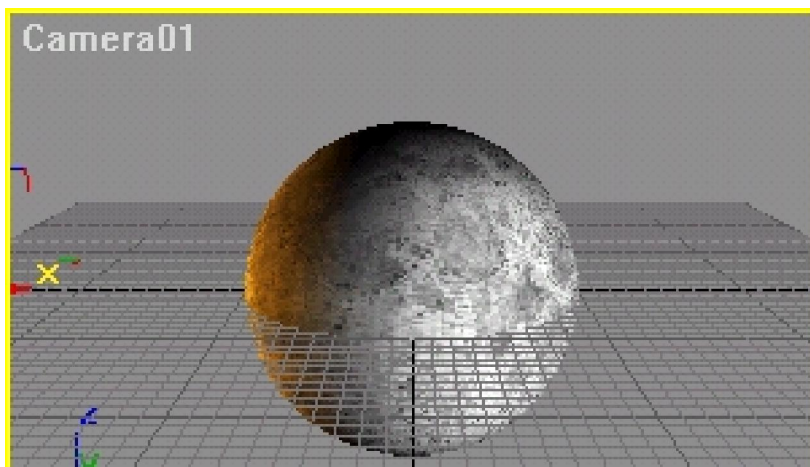
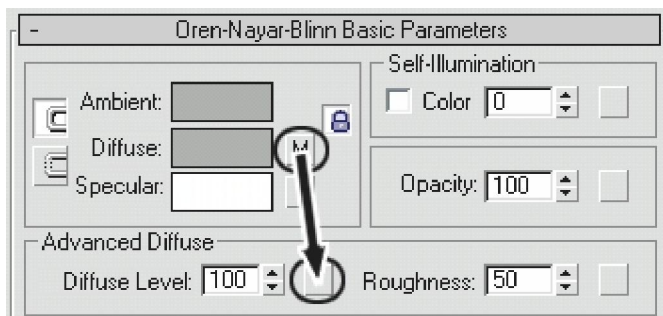
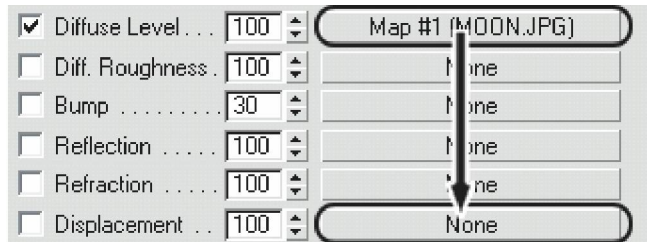
⑫ fill light의 색을 귤색으로 바꾼다. 그러면 화상에 온화한감이 생긴다.

⑬ Camera창문을 활성화하고 F9건을 눌러 시각화해본다.

2) 달의 기초모형에 재질을 입혀 실제의 달과 같이 만든다.

① 재질편성기를 열고 첫번째 견본구를 선택하여 이름을 Earth's Moon으로 바꾼다.

② Shader Basic Parameters의 Shader목록에서 Oren-Nayar-Blinn을 선택한다. 그러면 빛을 받아 강조되는 부분이 어두워지면서 견본구가 보다 부드러워진다.

③ Diffuse선택4각형을 찰칵하고 Material/Map Browser에서 Bitmap를 선택한 다음 OK 단추를 찰칵한다.

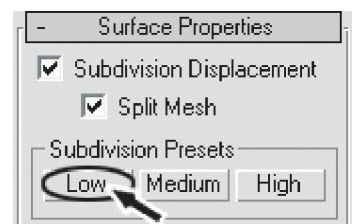
④ Select Bitmap Image File대화칸에서 MOON.JPG파일을 선택하고 Open단추를 찰칵한다. 그러면 달화상이 견본구 Earth's Moon에 나타난다.

⑤ 를 찰칵하여 우에서 만든 Earth's Moon재질을 달에 적용한다.

⑥ 를 찰칵하고 아래 그림과 같이 Diffuse선택4각형을 Diffuse Level선택4각형으로 끌어간다. 이때 나타나는 대화칸에서 Instance를 선택하고 OK단추를 찰칵한다.

⑦ Maps항목을 펼치고 아래 그림과 같이 Diffuse Level통로에 있는 Map선택단추를 Displacement통로에 있는 Map선택단추로 끌어간다.

이때 나타나는 대화칸에서 Copy를 선택하고 OK단추를 찰칵한다. 그러면 Displacement통로에 Map #2(MOON.JPG)가 생긴다.

Bump와 마찬가지로 Displacement도 물체의 겉면을 울퉁불퉁하게 해준다. 그러나 물체의 겉면을 바꾸지는 않는다.

Displacement의 Amount를 -20으로 설정한다.

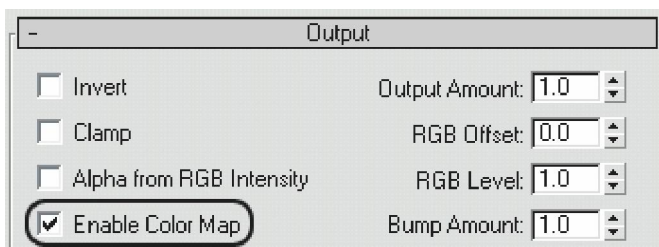
⑧ 창문에서 달을 선택하고 오른쪽찰칵한다. 이때 나타나는 지름차림표에서 TRANSFORM\Convert To\Editable Mesh를 실행한다.

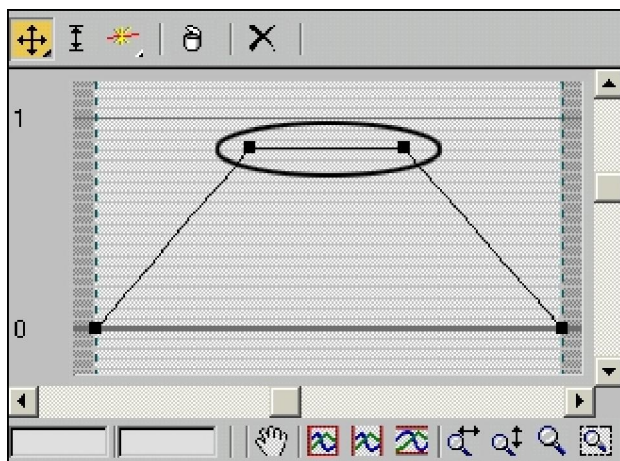
⑨ 변형판의 Surface Properties목록에서 Subdivi sion Displacement를 선택하고 Low단추를 찰칵한다. 이렇게 하면 달겉면이 지나치게 복잡해지는것을 막는다.

F9건을 눌러 시각화해보면 달의 겉면이 울퉁불퉁하게 나타난다.

⑩ Displacement의 Amount를 -50으로 설정한다.

⑪ Maps목록에서 Displacement통로의 Map단추를 찰칵한다.

⑫ Bitmap Parameters목록의 Cropping/Placement항목에서 View Image를 찰칵한다. 화상에서 보게 되는 회색값에 따라 겉면의 변화를 조절한다. 화상을 닫는다.

⑬ Output목록을 펼치고 Enable Color Map를 선택한다.

⑭ Scale DeFormation(X)대화칸을 펼치고 흘림띠를 끌어서 그라프와 그라디엔트띠가 보이게 한다. 그리고 오른쪽에 있는 점을 수직으로 이동시켜 첫 점과 수평으로 놓이게 한다. 그러면 그라디엔트띠가 검은색으로 된다.

⑮ (Add Point: 점을 추가)를 찰칵하여 1/3, 2/3되는 위치에 점들을 추가한다.

⑯ (Move)를 찰칵하고 앞에서 만든 두개의 점을 웃쪽으로 이동시켜 그라프가 제형 모양으로 되게 한다. 그러면 화상의 회색부분에서 울퉁불퉁해진다.

⑰ Displacement Amount를 20으로 설정한다. 시각화해보면 분화구가 있는 달이 얻어 진다.

2. 하늘에 있는 별만들기

Environment(주위환경)의 Background(배경)에 Noise Map를 적용하여 별이 총총한 밤 하늘을 만들어본다.

1) 별만들기

① 차림표띠에서 Rendering\Environment를 실행한다.

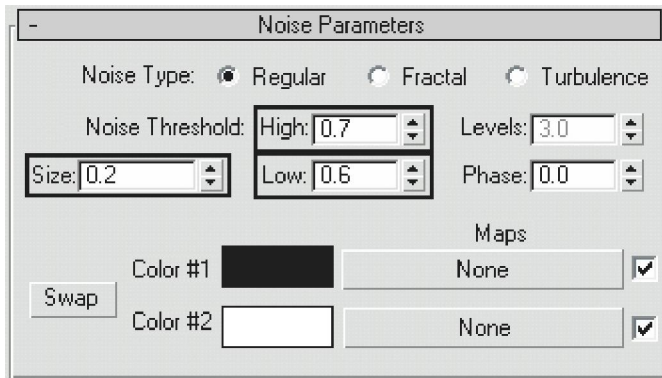
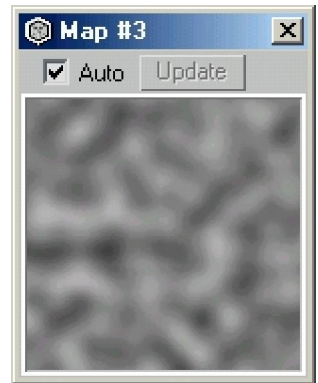
② Environment대화칸의 Common Parameters목록에서 Environment Map선택단추를 찰칵한다.

③ Material/Map Browser에서 Noise를 선택하고 OK단추를 찰칵한다. 그러면 Environment Map에 Map #3(Noise)이 나타난다.

④ 재질편성기를 연다.

⑤ Environment대화칸에서 Map #3(Noise)을 마우스로 끌어서 재질편성기의 새로운 견본구에 가져간다. 이때 나타나는 대화칸에서 Instance를 선택하고 OK단추를 찰칵한다.

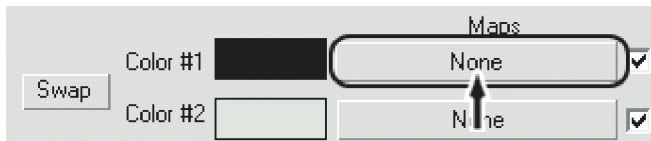
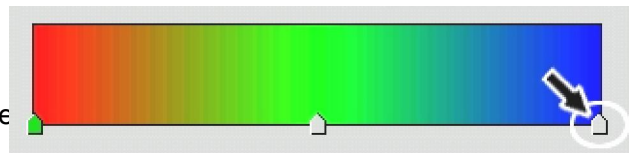
그러면 재질편성기에서 견본구가 없어지고 그 자리에 그림과 같이 Noise가 나타나는데 Noise는 Map이므로 4각형으로 나타난다.

⑥ Noise Parameters목록에서 아래와 같이 설정한다.

Noise Threshold의 High(최고)와 Low(최저)를 비슷하게 설정하면 흰색과 검은색사이의 범위가 좁아지면서 Noise는 점으로 나타난다.

⑦ Camera창문을 활성화하고 시각화해보면 많은 별같은것들이 하늘에 나타난다.

⑧ Low를 0.65로 증가시켜 별들의 개수를 줄인다. 그리고 Color #2를 연한 회색으로 변경시켜 별들의 밝기를 약하게 한다.

2) 은하수만들기

① Noise Parameters목록에서 Color #1선택단추를 찰칵한다.

② Material/Map Browser에서 Gradie

③ Gradient Ramp Parameters목록을 펼친다.

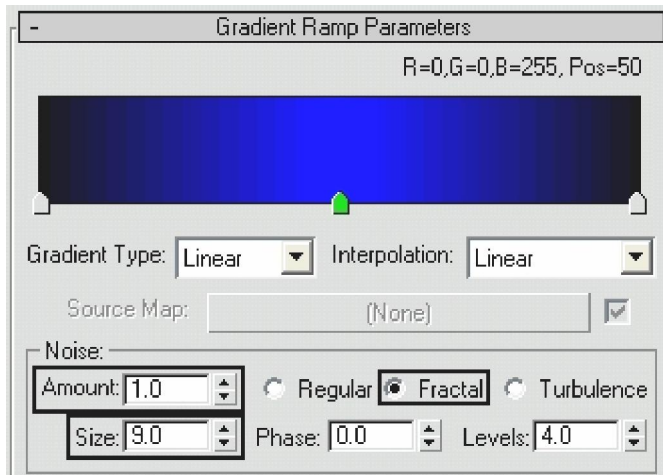
④ 그라디엔트표시부분의 오른쪽에 있는 기발을 두번찰칵하여 Color Selector대화칸을 펼친다.
검은색으로 설정한다.

⑤ Color Selector대화칸을 닫지 않은 상태에서 왼쪽에 있는 기발을 찰칵하여 그것도

검은색으로 설정한다. 가운데기발을 찰칵하여 파란색으로 설정하고 Color Selector대화칸을

닫는다.

⑥ Noise목록의 On을 찰칵하여 선택한다.

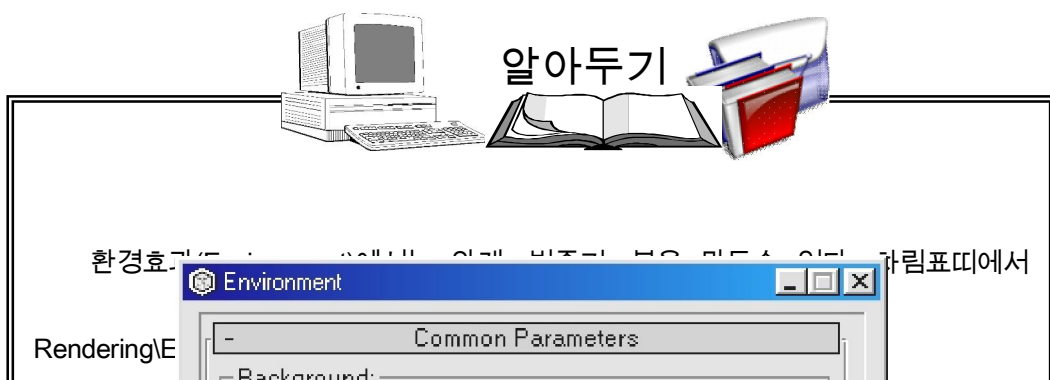
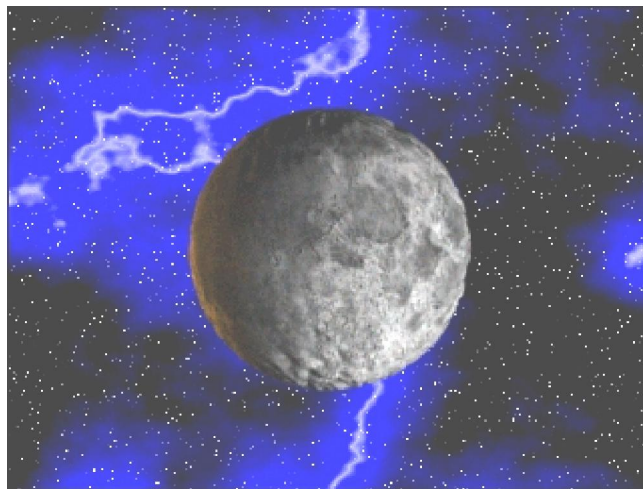
⑦ Gradient Ramp Parameters목록의 Noise항목을 다음 그림과 같이 설정한다.

⑧ 장면을 시각화한다. 그러면 하늘가에 여기저기 퍼져나간 파란색의 별구름이 나타난다. 만일 배경이 제대로 나오지 않으면 Coordinates목록에서 Environ이 선택되여있는가를 확인한다.

⑨ 보다 극적인 효과를 내기 위하여 별구름에 《번개》를 만들어준다.

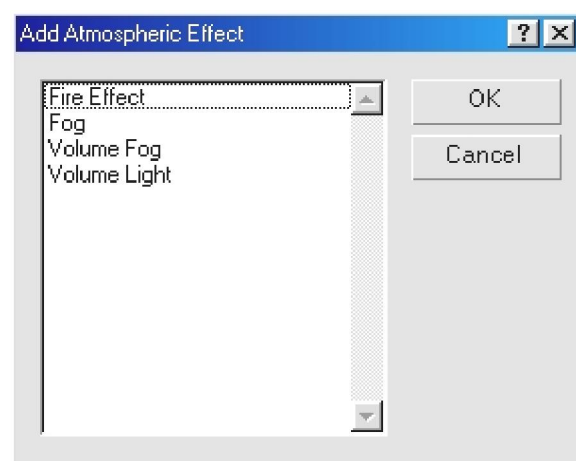
Gradient Ramp Parameters목록에서 그라디엔트막대기의 가운데기발의 오른쪽과 왼쪽을 찰칵하여 기발을 2개 더 만든다. 기발이 너무 많이 생겼으면 오른쪽찰칵하고 Delete지령을 실행하여 없앤다.

⑩ 가운데기발을 두번찰칵하여 연푸른색으로 설정한다.

⑪ Gradient Ramp Parameters목록의 Noise항목에서 Levels를 6으로 설정한다.

이렇게 하여 달이 완성되였다. 결과화상은 다음과 같다.

Level은 Light의 Multiplier와 같은 기능이다. Tint와 Level은 Ambient에는 영향을 미치지 않는다.

Ambient는 빛을 받지 않는 부분의 색을 조절한다. 재질편성기에서 나오는

Add Atmospheric Effect
Fire Effect
Fog
Volume Fog
Volume Light
OK
Cancel

3. 달의 운동만들기

① 동화조종판에서 Animate단추를 찰칵하면 Animate단추가 빨간색으로 된다.

모형의 움직임을 설정할수 있게 된다.

② 작업창문에서 달을 선택한다. 도구띠에서 를 선택하고 달을 움직여 달이 처음에 화면의 왼쪽아래에 놓이도록 한다.

③ 시간미끄럼대를 100프레임으로 옮긴다.

④ 도구띠에서 를 선택하고 달의 위치를 화면의 오른쪽우로 옮긴다.

⑤ 도구띠에서 를 선택하고 달의 크기를 작게 변화시킨다.

⑥ Animate단추를 다시 찰칵한다. 그러면 Animate단추의 빨간색이 없어진다.

⑦ 동화조종판에서 (Play: 재생)단추를 찰칵한다.

그러면 달이 이동하면서 크기가 달라지는것을 볼수 있다.

3ds max에서는 모형을 선택하고 그의 위치와 크기를 변화시킨 다음에 시간미끄럼대를 이동시키고 값을 입력하는 방법으로 모형의 움직임을 만들 … 프로 주어서 모형의 움직임을 만들수도 있다.

우선 화면아래쪽에 있는 Animate단추를 찰칵한다.

그러면 Animate단추가 붉은색으로 표시된다.

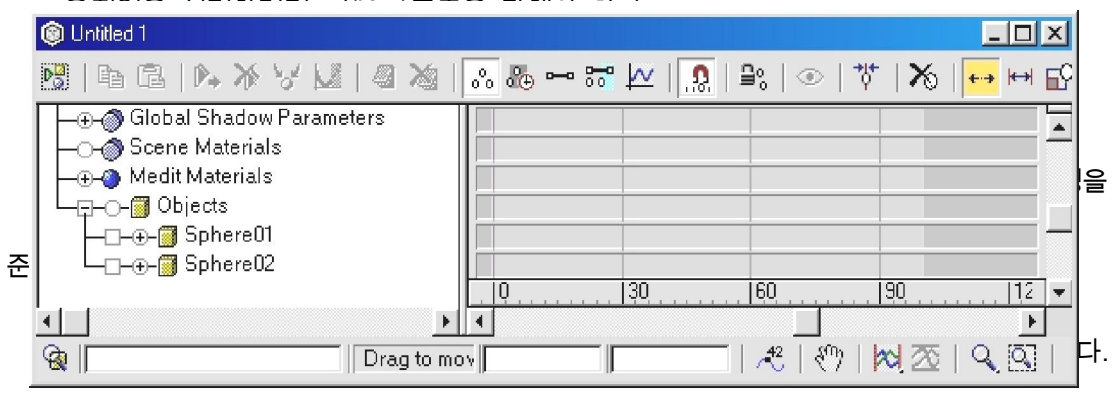
이제부터 모형의 움직임을 설정할수 있다.

움직임의 설정은 창문아래에 있는 시간미끄럼대에서 진행한다.

시간미끄럼대의 아래에 놓인 자리길띠의 눈금은 프레임을 나타낸다.

…을

준 …

…다.

여기서는 움직임을 그라프로 나타낸다. 즉 움직임의 경로를 이동, 회전, 확대축소별로, 축별로 지정하여 좀 더 세밀하게 조종할수 있다.

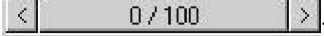
창문의 왼쪽에는 이동, 확대축소, 회전중에서 어떤 변화를 나타내며 어느 축에서 하겠는가를 선택한다. 오른쪽에서는 우에 있는 그라프편집도구로 움직임을 그림으로 표시한다.

련습문제

1. 조명을 만들어 모형에 빛을 비쳐보아라.
2. 조명의 색갈을 변화시키면서 빛을 비쳐보아라.
3. Cast Shadow를 리용하여 그림자를 만들어보아라.
4. Multiplier값을 변화시키면서 빛을 비쳐보아라.
5. 그림자에 화상을 적용해보아라.
6. 조명에 Volume Light효과를 적용해보아라.
7. 조명에 Lens Effects효과를 적용해보아라.
8. Camera를 움직이게 하여 모형의 움직임효과를 만들어보아라.
9. Depth of Field를 적용해보아라.
10. Motion Blur를 적용해보아라.
11. 시간에 따라 모형의 색갈이 변하는 움직임효과를 만들어보아라.
12. Self-Illumination을 리용하여 모형이 발광하게 해보아라.
13. 조명을 만들어 Ambient색을 적용해보아라.
14. 구에 자기이름을 Mapping해보아라.
15. 병에 상표를 붙여보아라.
16. 앞표지와 뒤표지, 옆표지를 가진 사전을 하나 만들어보아라.

실 습

소행성이 레이자빛을 받아 폭파되는 동화상을 만들어보기로 하자.

작업을 네 단계로 나누어 진행하기로 한다.

첫번째 단계에서는 레이자빛이 소행성에로 가는 움직임을 만들고 두번째 단계에서는 소행성이 폭파되는 장면을 만든다. 세번째 단계에서는 소행성이 없어지면서 파편쪼각들이 날아가는 움직임을 만들고 네번째 단계에서는 여러가지 효과들을 주어 동화상을 진실감이 나게 완성한다.

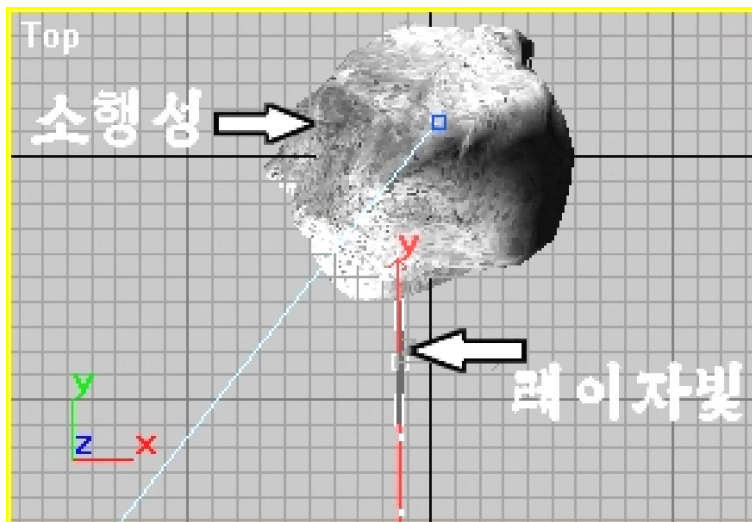
1) 첫번째 단계

◇ 레이자빛과 소행성에 움직임주기

① exploding_asteroid1.max파일을 연다. 이 파일에는 소행성과 laser blast(레이자폭파)라는 이름을 가진 얇은 직6면체가 만들어져있다.

② laser blast를 선택한다.

③ 동화조종판에서 Animate단추를 찰칵하고 시간미끄럼대 0 / 100 를 10프레임으로 옮긴다.

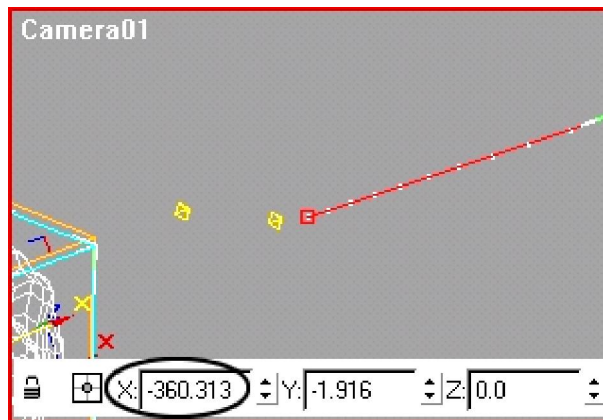
④ Top창문에서 laser blast를 Y축을 따라 이동시켜 소행성에 닿게 한다.

⑤ 시간미끄럼대를 0프레임으로 옮긴다.

⑥ 소행성의 X자리표값을 대략 -360정도로 설정한다. 그러면 소행성이 Camera창문의 제일 끝에 놓이게 된다.

⑦ 시간미끄럼대를 10프레임으로 옮긴다.

⑧ 소행성이 레이자빛과 부딪치도록 위치를 옮기고 Y, Z축주위로 회전시킨다.

⑨ Animate단추를 다시 찰칵한다.

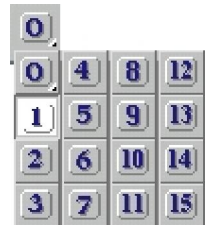
⑩ Camera창문에서 동화상을 재생해보면 레이자빛이 소행성에 가서 멎는다. 창문에서 보이는 점선은 레이자빛의 자리길을 나타내는 선이다.

◇ **레이자빛에 재질입히기**

① 재질편성기를 연다.

② 재질편성기에서 새로운 견본구를 선택하고 이름을 laserblast material로 바꾼다.

③ Blinn Basic Parameters목록에서 Color를 찰칵하여 해제하고 Color값을 100으로 설정한다.

④ Diffuse색을 R=255, G=114, B=0으로 설정한다.

⑤ 재질편성기의 도구띠에서 0 을 찰칵하고 1을 선택한다. 이것은 후에 발광효과를 주기 위해서이다.

⑥ 만든 재질을 레이자빛에 적용한다.

⑦ 재질편성기를 닫는다.

2) 두번째 단계

이제는 소행성이 폭파되는 장면을 만들어야 한다.

3ds max로 폭파장면을 만드는 방법에는 여러가지가 있다.

여기서는 소행성이 어떤 평면에 놓이게 하고 그 평면에 효과를 주어 화면에서는 소행성이 폭파되는것처럼 보이도록 하는 방법으로 폭파장면을 만들어보기로 한다.

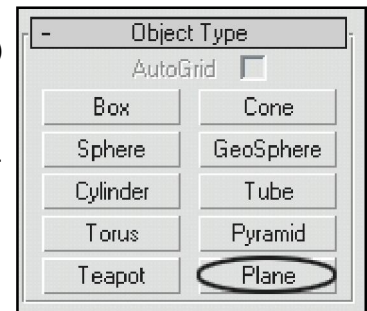
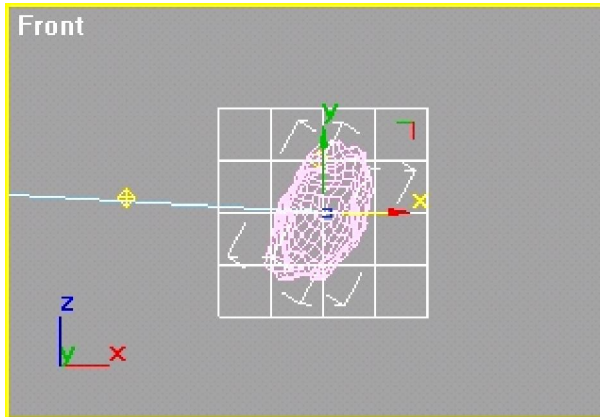
◇ **평면만들기**

① 시간미끄럼대를 10프레임으로 옮긴다.

② Front(정면)창문을 활성화하고 건반에서 W건을 눌러 확대한다.

③ 작업창문조종판에 있는 단추를 리용하여 소행성이 창문의 중심에 놓이게 이동시킨다.

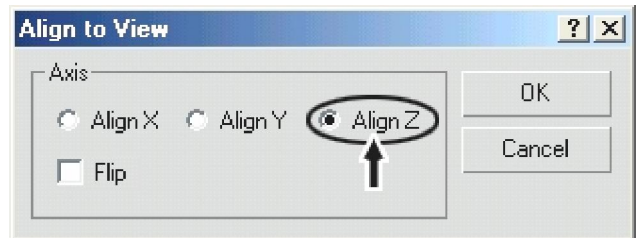
④ 창작판의 를 찰칵하고 모형종류에서 Plane(평면)을 선택한다.

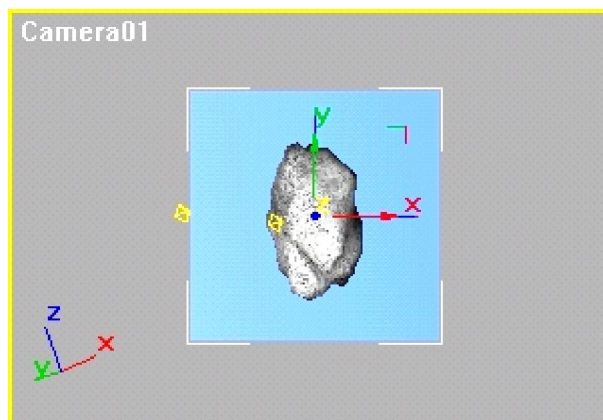
⑤ Front창문에서 소행성이 포함되게 Plane객체를 만든다.

⑥ Plane객체의 이름을 Explosion Plane01로 바꾼다.

⑦ Plane을 선택하고 변형판으로 가서 Length(길이)와 Width(너비)를 300으로, Length Segs(길이토막수)와 Width Segs(너비토막수)를 1로 설정한다.

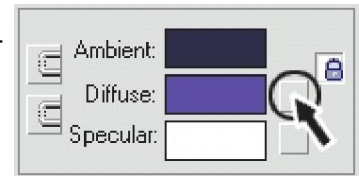
⑧ 건반에서 W건을 눌러 4개 창문이 보이게 한다.

⑨ Top창문에서 Plane객체를 선택하고 Y축방향으로 이동시켜 소행성의 뒤쪽에 배치한다.

⑩ Camera창문을 활성화하고 Plane이 선택된 상태에서 Tools\Align to View를 실행하여 Align to View대화칸을 펼친다.

⑪ Align to View대화칸에서 Align Z가 선택되여있는가를 확인하고 OK단추를 찰칵한다.

그러면 Camera창문에서 Plane이 정렬된다. 즉 Plane의 수직선이 카메라방향과 일치하게 된다.

◇ **평면에 재질입히기**

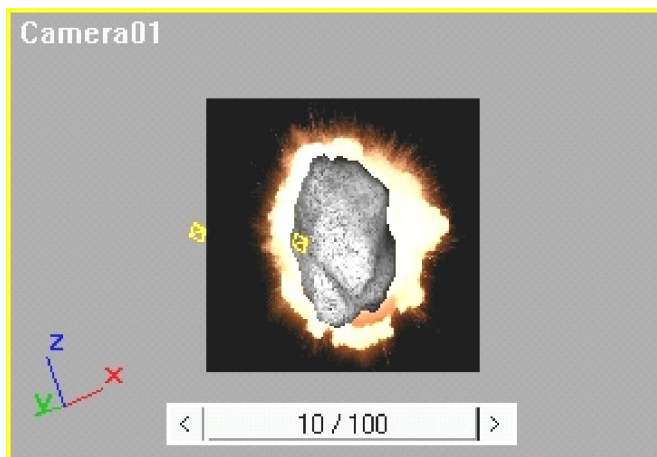
① 재질편성기를 열고 새로운 견본구를 하나 선택하고 이름을 Material Explosion (폭파재질)으로 바꾼다.

② Blinn Basic Parameters항목에서 Diffuse선택4각형을 찰칵한다.

③ Material/Map Browser대화칸에서 Bitmap를 선택하고 OK단추를 찰칵하면 Select Bitmap Image File대화칸이 열린다.

④ 대화칸에서 Tutorials\Obj_Design\Hercules.avi파일을 선택하고 Open단추를 찰칵한다.(이것은 흑백색으로 된 동화상이다.) 그러면 동화상이 Duffuse Map로 된다.

⑤ 이렇게 만든 Material Explosion재질을 Plane에 적용한다.

⑥ 재질편성기의 도구띠에서 를 찰칵한다.

그러면 시간미끄럼대가 10프레임에 있기때문에 창문에 동화상의 10번째 프레임이 나타나는것을 볼수 있다.

시간미끄럼대를 옮기면 그에 맞는 Diffuse Map가 표시된다.

이렇게 하여 Plane에 폭파모양이 나타났다. 이제는 Plane이 보이지 않게 하여 진짜 폭파되는것처럼 보이게 한다.

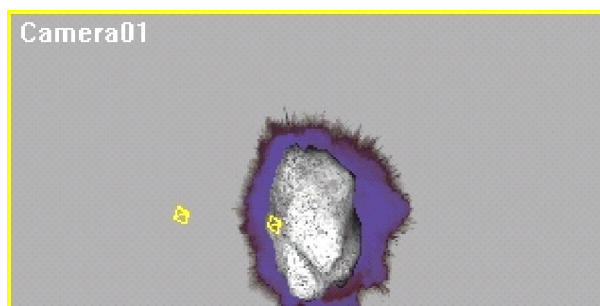
⑦ 재질편성기의 도구띠에서 를 찰칵한다.

⑧ Maps항목을 펼치고 Opacity통로의 Map선택단추를 찰칵한다.

⑨ Material/Map Browser대화칸에서 Bitmap를 선택하고 OK단추를 찰칵한다.

⑩ 펼쳐지는 Select Bitmap Image File대화칸에서 Tutorials\Obj_Design\Hercu les.avi파일을 선택하고 OK단추를 찰칵한다.

⑪ 를 찰칵하고 Maps항목을 닫는다.

⑫ 를 찰칵한다. 그러면 폭파부분을 제외한 나머지부분이 보이지 않게 된다.

⑬ Blinn Basic Parameters목록의 Specular Highlights항목에서 Specular Level과 Glossiness를 0으로 설정한다.

이렇게 하여 Plane이 빛을 받아 반사하는 현상을 없앤다.

⑭ Plane을 선택하고 오른쪽찰칵하여 Properties를 선택한다.

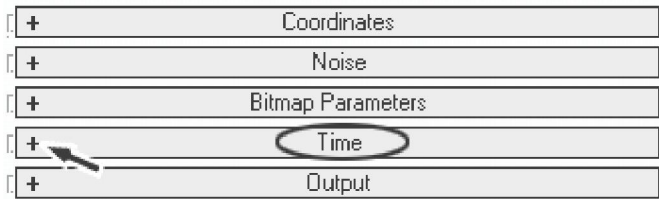
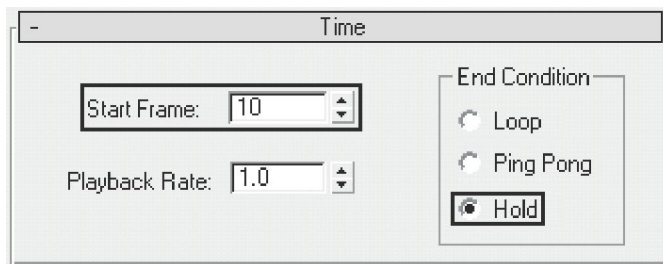
⑮ 펼쳐지는 Object Properties(모형속성)대화칸의 Rendering Control(시각화조종체)항목에서 Cast Shadow와 Receive Shadow를 해제하고 OK단추를 찰칵한다.

◇ **시간맞추기**

동화상을 재생해보면 알수 있는바와 같이 폭파장면은 레이자빛이 소행성에 부딪치는 순간 즉 10프레임부터 시작되지 않는다.

① 재질편성기의 Maps항목에서 Diffuse Map단추를 찰칵한다. 현재 Diffuse Map는

Hercules.avi이다.

② Time목록을 펼친다.

③ Start Frame(시작프레임)을 10, End Condition(끝상태)항목에서 Hold(포착)를 선택한다.

④ 재질편성기의 도구띠에서 (Go Forward to Sibling)를 찰칵하여 다음 Map를 본다. 이 경우에는 Opacity Map가 된다.

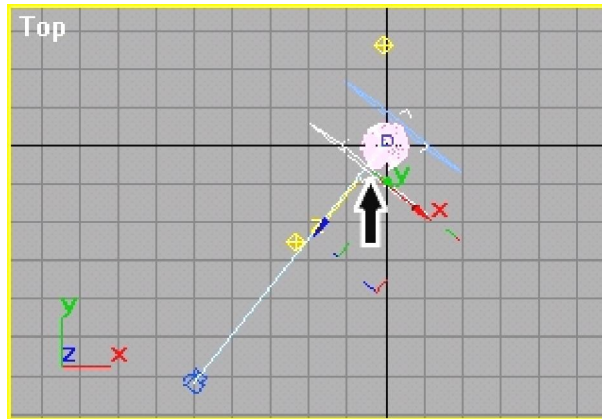
⑤ ② - ④의 공정을 Hercules.avi에 대하여 반복한다.

⑥ 재질편성기를 닫는다.

⑦ 이제는 Explosion Plane01을 복사하여 그것을 소행성의 앞에 배치하기 위하여Top 창문에서 Explosion Plane01을 선택한다. 그리고 건반에서 공백건을 눌러 선택을 고정한다.

⑧ 도구띠에서 를 찰칵한다.

⑨ 도구띠의 (Reference Coordinate System: 자리표계참조)에서 Local을 선택한다. 그러면 Plane을 수직선방향으로 이동시킬수 있다.

⑩ 건반에서 Shift건을 누르면서 마우스를 Z축방향으로 끌기하여 Explosion Plane 01의 복사본을 만든다.

복사된 Plane이 소행성의 앞에 놓이게 한다.

⑪ 두 Plane객체를 선택하고 오른쪽찰칵하여 나타나는 지름차림표에서 Hide Selection(선택모형숨기기)을 선택한다. 그러면 Plane들이 창문에서 없어진다.

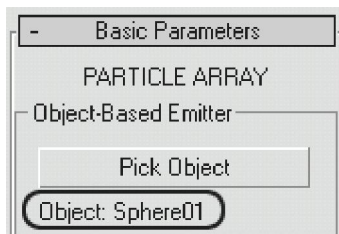
3) 세번째 단계

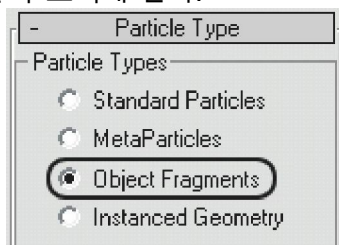
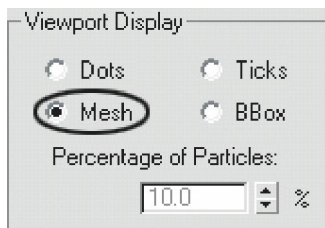
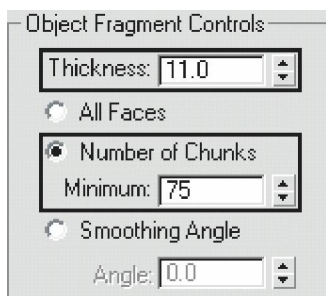
이제는 연기, 구름, 물보라, 폭파, 비 등과 같은 립자분산현상을 표현하는 립자체계(Particle System)를 리용하여 소행성이 폭발하게 한다.

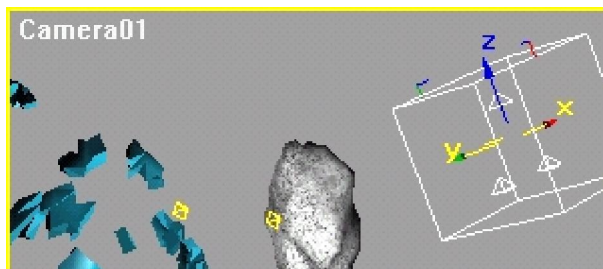
◇ **립자체계만들기**

① 창작판의 에서 Particle Systems를 선택하여 나타나는 모형종류에서 PArray를 선택한다. Parray는 분수와 같은 립자분출현상이나 폭발현상을 만드는것이다.

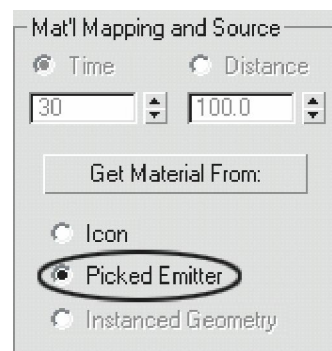
② 임의의 창문에서 마우스를 끌기하여 Parray모형을 만든다.

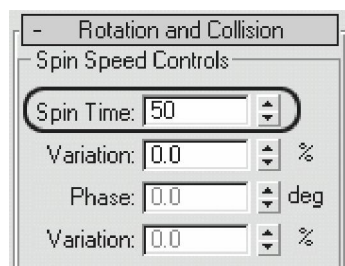
③ Basic Parameters목록에서 Pick Object(모형선택)를 찰칵한다. 그리고 창문에서 소행성을 찰칵한다. 소행성의 이름인 Shpere01이 Pick Object단추밑에 표시된다. 이것은 립자체계가 소행성과 련결되였다는것을 보여준다.

창문과 지령판의 경계선에 마우스지시자를 가져가면 마우스지시자의 모양이 ↔로 바뀐다. 이때 마우스를 끌기하여 지령판을 확장하여 모든 항목들이 보이게 한다.

④ Particle Type(립자종류)목록을 펼친다. Object Fragments(쪼각모형)를 선택한다.

⑤ Object Fragment Controls목록이 나타나게 마우스를 끌기하고 다음 그림과 같이 설정한다.

⑥ Basic Parameters목록의 Viewport Display항목에서 Mesh를 선택한다.(오른쪽웃그림)

시간미끄럼대를 이동시켜본다. 그러면 소행성쪼각이 화면에 나타난다.

◇ **재질입히기**

① Particle Type목록의 Material Mapping and Source 항목에서 Picked Emitter를 선택한다.

② Get Material From단추를 찰칵한다. 그러면 립자에 소행성의 재질이 입혀진다.

◇ **립자들의 움직임조종하기**

① Rotation and Collision(회전 및 충돌)목록을 펼친다.

② Spin Speed Controls항목에서 Spin Time을 50으

로 설정한다. 이렇게 하면 파편쪼각들이 회전한다.

③ Object Motion Inheritance목록을 펼치고 Infl uence

를 0으로 설정한다.

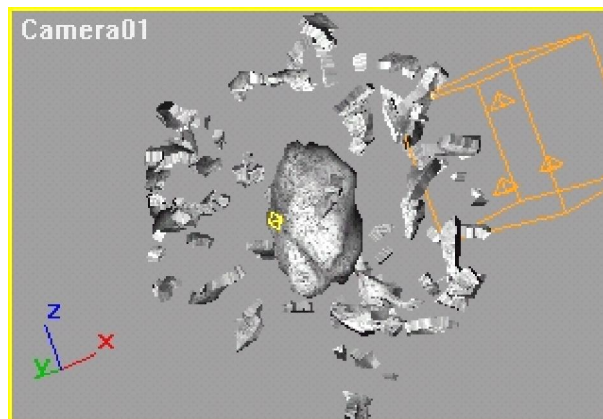
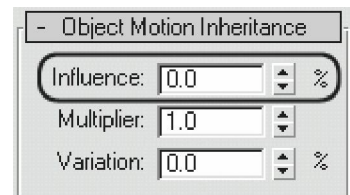
이렇게 하면 립자들의 움직임이 소행성의 움직임과

회전에 영향을 미치지 않는다.

한편 립자들은 0프레임으로부터 발생한다. 그런데 소행성은 10프레임부터 폭파되므로 립자들이 발생하는 시간을 조절해주어야 한다.

④ Particle Generation(립자발생)목록을 펼친다.

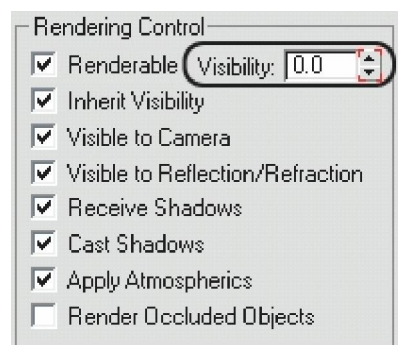
Particle Time(립자시간)항목에서 Emit Start(방출시작)를 11, Life(수명)를 89로 설정한다. 그러면 립자들이 마지막 프레임까지 존재하게 된다.

⑤ 시간미끄럼대를 옮기면서 창문에서 립자효과가 제대로 나타나는가를 본다.

◇ **소행성없애기**

폭파장면을 보다 실감있게 하자면 폭파될 때 소행성이 없어져야 한다. 여기서는 소행성에 레이자빛이 쪼여졌을 때 소행성이 없어지도록 한다.

① 시간미끄럼대를 11프레임으로 옮기고 Animate단추를 찰칵한다.

② 소행성을 오른쪽찰칵하고 Properties를 선택한다.

만일 립자들때문에 소행성을 선택하기 힘들면 건반에서 H건을 눌러 Select Objects대화칸을 열고 Sphere01을 선택하면 된다.

③ Object Properties대화칸의 Rendering Control목록에서 Visibility값을 0으로 설정한다.

그러면 수값조절단추에 붉은색테두리가 생긴다. 이것은 그 값이 동화화되여있다는 표시이다.

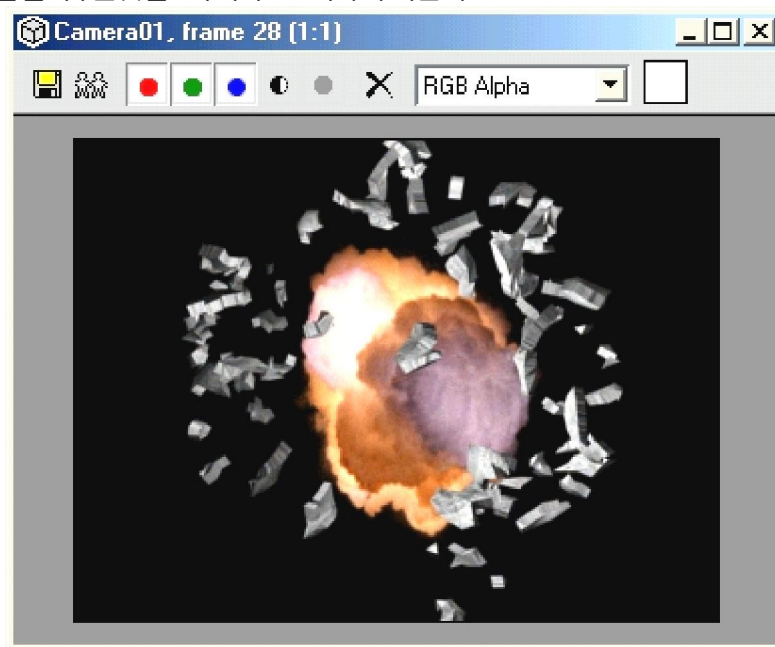
④ OK단추를 찰칵하여 Object Properties대화칸을 닫는다.

⑤ Animate단추를 다시 찰칵하여 Animation방식에서 탈퇴한다.

⑥ 시간미끄럼대밑에 있는 눈금이 새겨진 자리길띠(Track Bar)의 0번프레임을 오른쪽찰칵하여 나타나는 지름차림표에서 Sphere01: Visibility를 선택한다. 이때 펼쳐지는 Sphere01: Visibility대화칸의 Out항목의 모양을 계단형태로 바꾼다. 이렇게 하면 소행성은 다음 키프레임까지 보이게 된다.

⑦ 시간미끄럼대를 10프레임으로 옮긴다.

⑧ laser blast를 선택하고 ② - ⑦의 공정을 반복하여 레이자빛이 소행성보다 한 프레임 먼저 없어지게 한다.

⑨ Plane들을 숨긴것을 해제하고 시각화해본다.

소행성이 폭파되면서 없어진다.

4) 네번째 단계

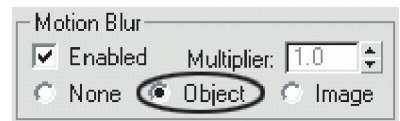
◇ Motion Blur추가하기

파편쪼각들이 빠른 속도로 움직이는것과 같이 보이도록 하자면 Motion Blur효과를 추

가해주어야 한다.

① 소행성을 선택하고 오른쪽찰칵하여 Properties를 선택한다.

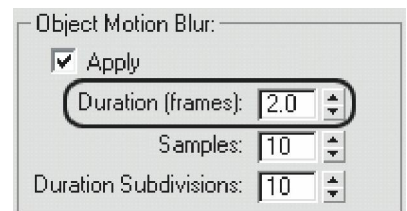
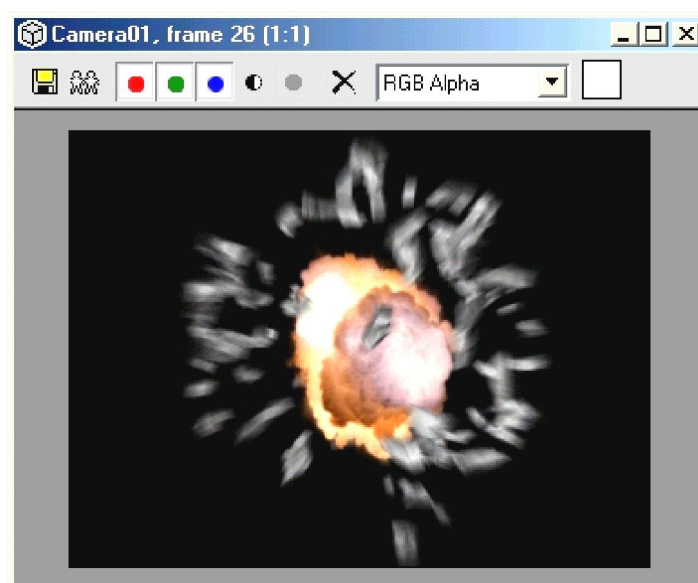
② Motion Blur항목에서 Object를 선택한다.

그러면 소행성에 Motion Blur효과가 적용된다.

③ PArray에도 Motion Blur효과를 준다.

④ 시간미끄럼대를 0프레임이 아닌 임의의 곳에 가져간다.

⑤ Camera창문을 활성화하고 도구띠에서 Render Scene을 찰칵한다.

⑥ Render Scene대화칸에서 Common Parameters목록을 닫고 Max Default Scanline A-Buffer목록을 펼친다.

Object Motion Blur항목에서 Duration(frames)을 2로 설정한다. 이렇게 하면 Motion Blur에 리용되는 프레임수가 늘어나 효과를 더 크게 해준다.

⑦ Render Scene대화칸의 밑부분에 있는 Render단추를 찰칵하여 시각화해본다.

◇ **조명추가하기**

소행성안에 조명을 설치하여 파편쪼각들이 빛을 뿜게 하면 보다 실감있는 효과를 만든다.

① 시간미끄럼대를 0프레임으로 옮긴다.

② 창작판의 [icon]를 찰칵하고 모형종류에서 Omni를 선택한다.

③ 창문에서 소행성의 중심을 찰칵하여 Omni조명을 만든다.

④ 조명이름을 Inside Asteroid Light(소행성내부조명)로 바꾼다.

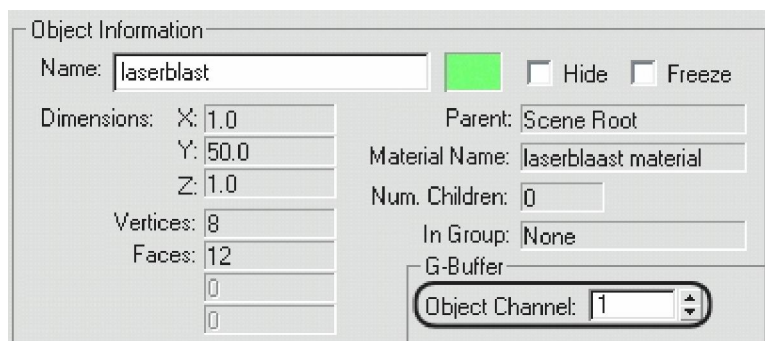
⑤ 조명의 색을 R=255, G=111, B=56으로 바꾼다.

⑥ 조명의 Multiplier를 4로 설정한다.

⑦ 도구띠의 [icon]를 리용하여 조명과 소행성을 련결시킨다. [icon]를 리용하여 두 물체를 련결시킬 때 처음물체가 자식물체로 되고 다음물체가 어미물체로 된다.

결과를 보고 알수 있는바와 같이 파편에 빛이 비쳐지는것을 볼수 있다.

◇ **레이자빛에 효과주기**

① 시간미끄럼대를 5프레임으로 옮긴다.
② laser blast를 선택하고 오른쪽찰칵하여 Properties를 선택한다.
③ Object Properties대화칸의 Object InFormation(모형정보)목록에서 G-Buffer의 Object Channel(모형통로)을 1로 설정하고 OK단추를 찰칵한다.

④ 차림표띠에서 Rendering\Effects를 실행한다.

⑤ Rendering Effects(시각화효과)대화칸의 Effects항목에서 Add단추를 찰칵하여 나타나는 대화칸에서 Lens Effects(렌즈효과)를 선택한 다음 OK단추를 찰칵한다.

⑥ 나타나는 Lens Effects Parameters(렌즈효과파라메터)목록에서 왼쪽칸의 Glow (빛내기)를 선택하고 >단추를 찰칵한다. 그러면 오른쪽칸에 Glow가 나타난다.

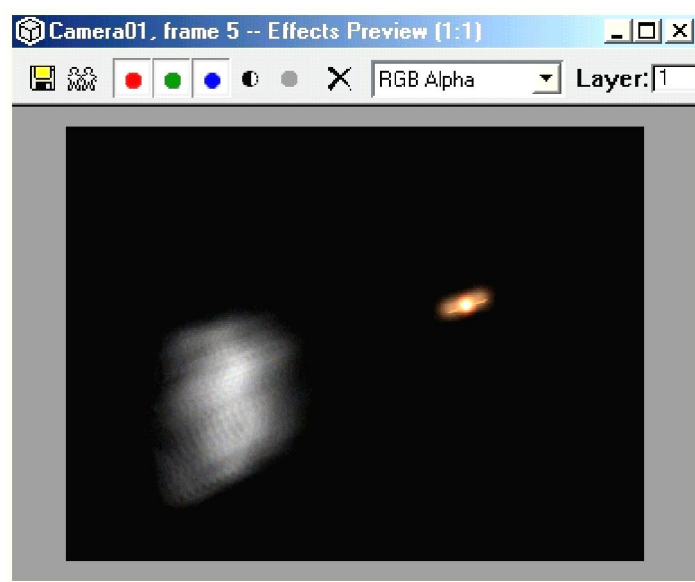
Glow Element항목의 Name칸에 laserglow라고 입력한다. Size를 5로 설정한다.

Radial Color항목에서 흰색4각형을 찰칵하여 R=235, G=120, B=60으로 바꾼다.

⑦ Options표쪽을 찰칵하고 Apply Element To항목에서 Image Centers를 선택한다. Image Sources항목에서 Object ID를 선택한다.

Lens Effects Globals항목의 Size를 5로 설정한다. Intensity를 400으로 설정한다.

⑧ Effects목록의 Preview항목에서 Update Scene(갱신된 장면)단추를 찰칵한다.

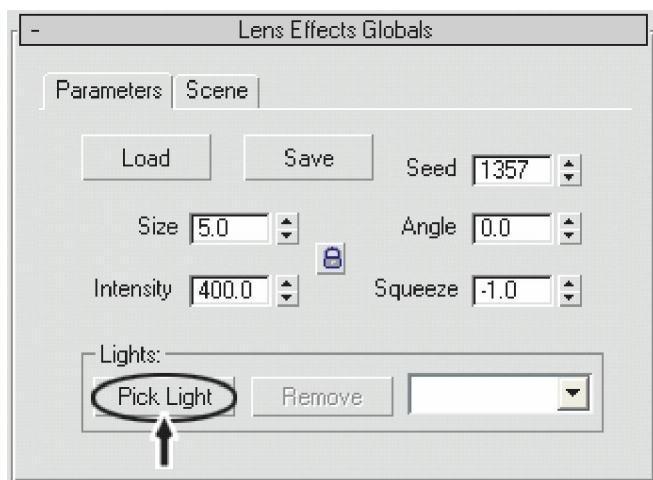
그러면 장면이 시각화된다.

Glow효과가 희미하게 적용된것을 알수 있다.

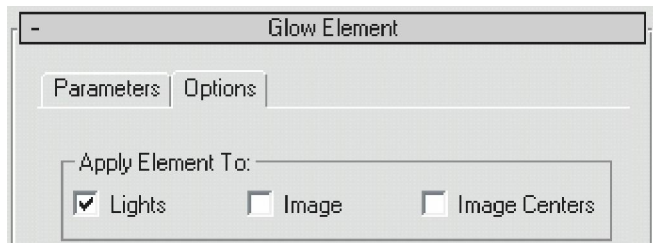
◇ **조명에 발광효과주기**

① 작업창문을 오른쪽찰칵하여 Unhide All(모두 드러내기)을 선택한다. 그러면 Explosion Plane들이 나타난다.

② Rendering Effects대화칸의 Lens Effects Parameters목록에서 왼쪽칸의 Glow를 다시 선택하고 >단추를 찰칵한다.

③ Glow Element항목에서 Name칸에 blast라고 입력한다.

④ Lens Effects Globals(전역렌즈효과)목록에서 Parameters표쪽을 선택하고 Pick Light(조명선택)단추를 찰칵한다.

⑤ 작업창문에서 Inside Asteroid Light를 선택한다.

⑥ Glow Element목록에서 Options표쪽을 찰칵하고 Apply Element To항목에서 Light를 선택한다.

⑦ Parameters표쪽을 찰칵하고 Occlusion을 0으로 설정한다.

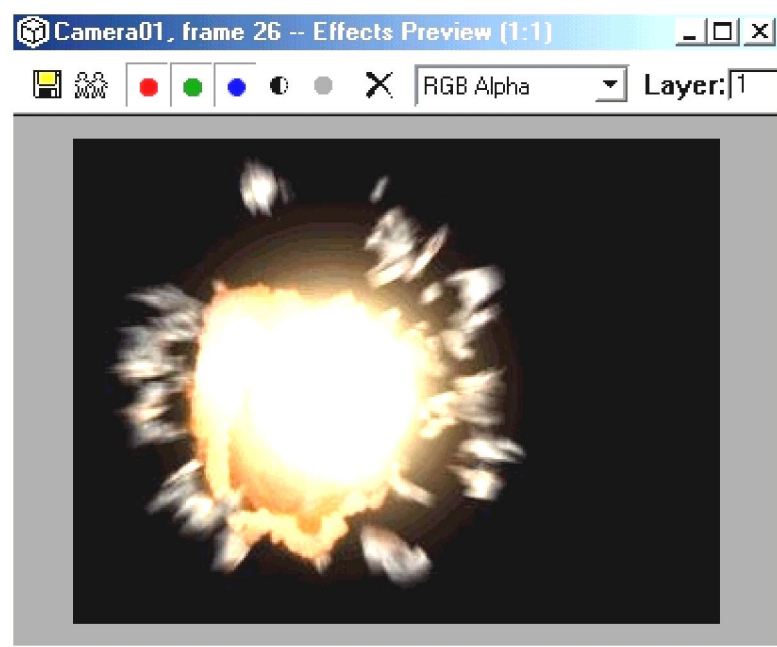
그러면 발광효과가 Explosion Plane을 뚫고 보이게 된다.

Glow Behind를 해제한다.

Radial Color항목에서 붉은색4각형(오른쪽에 있는)을 찰칵하여 R=242, G=150, B=0으로 설정한다.

Size를 300으로 설정한다.

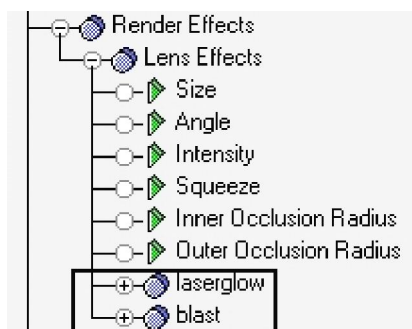
⑧ 시간미끄럼띠를 26프레임으로 옮긴다.

⑨ Effects목록의 Preview항목에서 Update Scene을 찰칵하여 시각화해본다.

Rendering Effects대화칸을 닫는다.

◇ **발광효과 동화화하기**

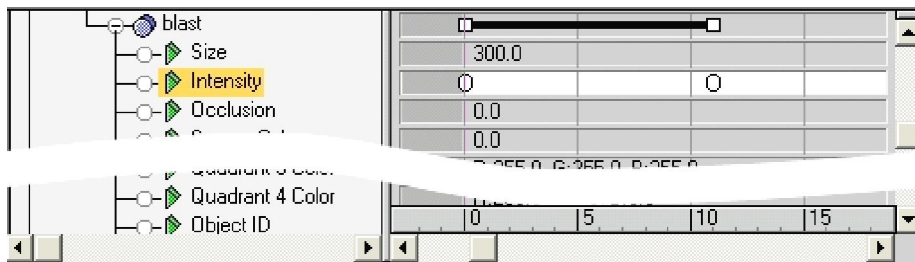
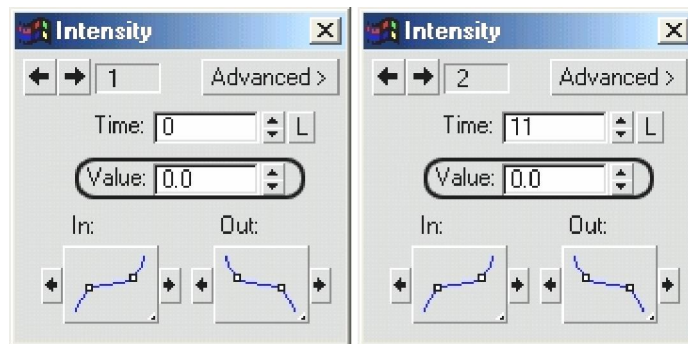
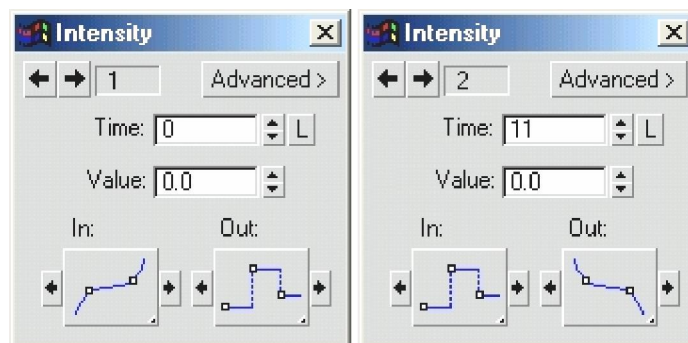
① 도구띠에서 [icon]를 찰칵하여 Track View대화칸을 펼친다.

② Track View대화칸의 왼쪽칸에서 Render Effects의 +기호를 찰칵한다.(왼쪽그림) 그러면 Render Effects가 펼쳐진다. 이때 +기호는 -기호로 바뀐다. 닫으려는 경우에는 –기호를 찰칵하면 된다.

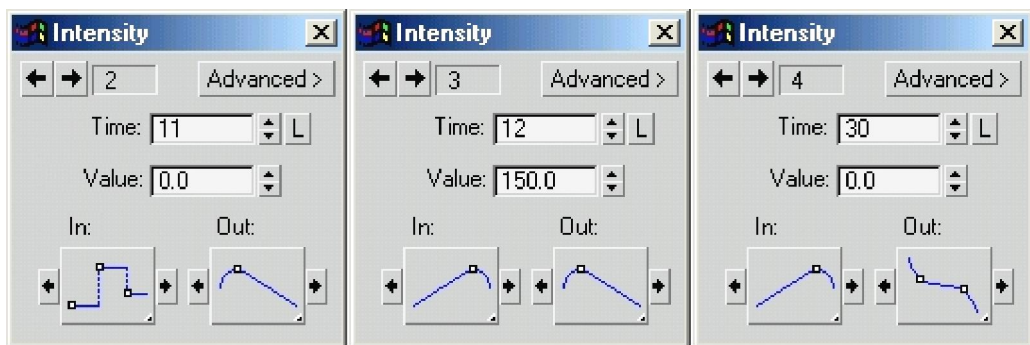
③ Lens Effects를 펼친다. 이미 만든 두개의 발광효과를 볼수 있다.(오른쪽그림)

④ blast를 펼친다.

Rendering Effects대화칸에 있는 발광효과의 속성이 여기에 다 표시되여있다. 여기서 발광효과를 동화화하기 위한 키를 설정할수 있다.

⑤ Intensity를 선택하고 Track View대화칸의 도구띠에서 [icon](Add Key: 키추가)를 찰칵한다.

⑥ 0번프레임과 11번프레임에 키를 설정한다.

⑦ 첫번째 키를 오른쪽찰칵하고 Value를 0으로 설정한다. 두번째 키의 Value도 0으로 설정한다.

⑧ 첫번째 키의 Out를 계단형태로 바꾼다. 그러면 두번째 키의 In도 계단형태로 바뀐다.

⑨ 12번프레임에 키를 또 설정하고 Value를 150으로 설정한다.

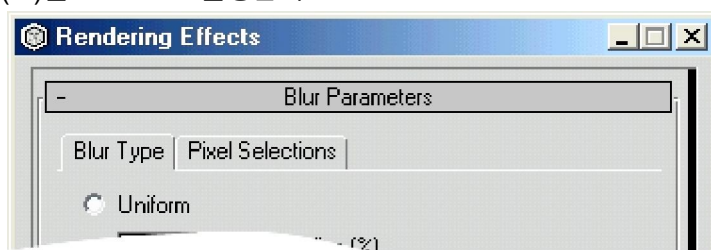
⑩ 30번프레임에 키를 또 설정하고 Value를 0으로 설정한다.

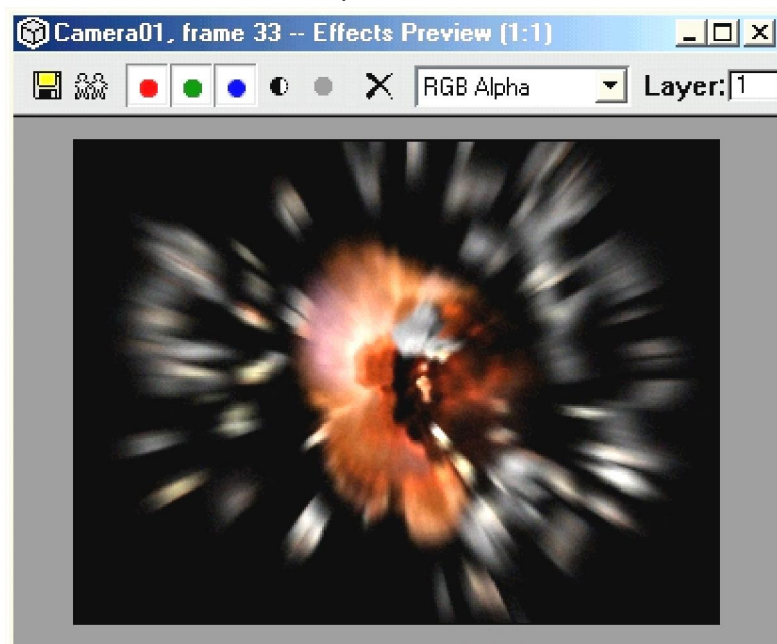
⑪ 11번프레임의 Out와 12번프레임의 In, Out 그리고 30번프레임의 In을 직선형태로 설정한다.

◇ **흐림효과주기**

① 차림표띠에서 Rendering\Effects를 실행한다.

② Rendering Effects대화칸의 Effects목록에서 Add단추를 찰칵하여 나타나는 대화칸에서 Blur를 선택하고 OK단추를 찰칵한다.

③ Blur Parameters목록의 Blur Type표쪽을 찰칵하고 Radial을 선택한다.

Pixel Radius(%)를 100으로 설정한다.

④ 시간미끄럼대를 33프레임으로 옮긴다.

⑤ Effects목록의 Preview항목에서 Update Scene을 찰칵하여 결과를 본다.

발광효과에서 Intensity에 키를 설정한것과 같은 방법으로 Radial Blur에도 키를 설정할 수 있다. 이렇게 하여 소행성의 폭파장면을 만들었다. 이제는 Camera창문을 활성화하고 동화상을 시각화하면 된다.

립자모형만들기

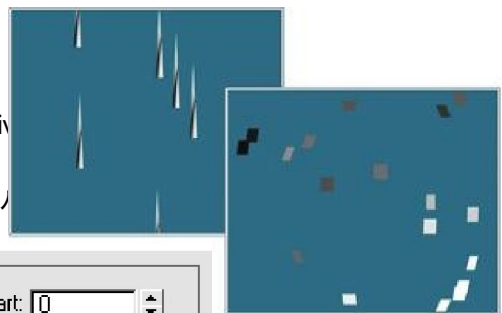
연기, 구름, 물보라, 폭파, 비 등과 같은 립자분산현상을 표현하는것이 립자체계(Particle System)이다. 립자체계를 구성하는 기본적인 요소는 다음의 3가지이다.

① 립자(Particle)

② 립자를 발생시키는 방출기(Emitter)

③ 립자에 작용하는 힘

창작판의 를 선택하고 Standard Primitiv

는 검은 3각형을 찰칵하여 나타나는 목록에

Systems

가 표시되

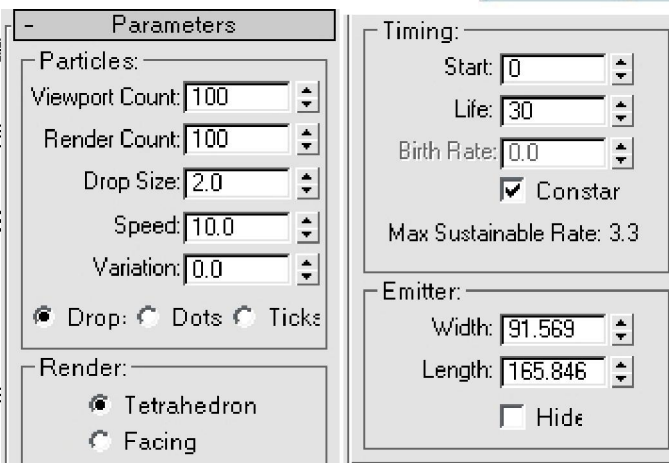
1) S

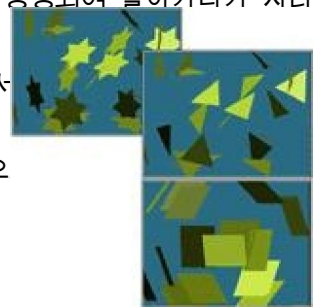
들어낸다

Spra

끌기하여 ... 다음 Play단추를 찰칵해보면 립

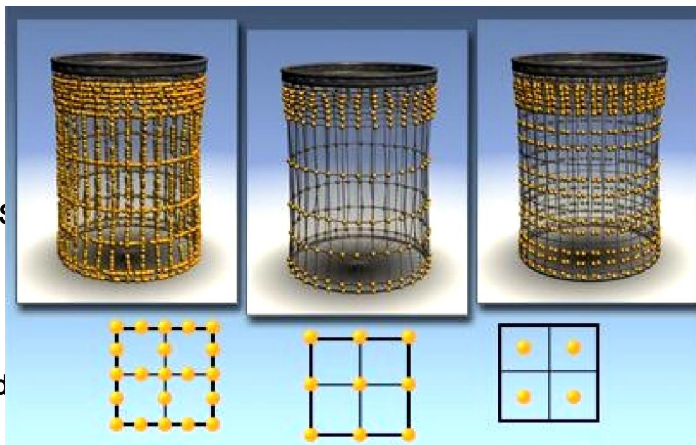
자들이 생성되는것을 볼수 있다. Parameters목록

을 보면 크게 네 부분으로 이루어져있다.

Render항목에서는 시각화할 때 립자들을 어떻게 나타내겠는가를 설정한다. Tetra hedron을 선택하면 Drop Size에서 설정한 크기의 3각뿔형태로 만들고 Facing을 선택하면 바른4각형으로 만든다.

Timing항목에서는 립자의 생성과 수명을 설정한다. Start에서는 립자가 나타나는 첫 프레임을 설정한다. Life에서는 개별적인 립자들이 방출기에서 생성되여 날아가다가 사라질 때까지의 수명을 프레임으로 설정한다. Birth Rate(생성률)에서 립자를 만들것인가를 설정한다. Constant를 선택하면 자동적으 다.

Emitter항목에서는 작업창문에서 방출기의 크기를 설정한다.

2) Snow - 다.

3) Super S 많이 사용된다.

4) Blizzard 많이 리용된다. 거의 모든 기능이 PArray와 류사하다.

5) PArray - Particle Array(립자배렬)는 두가지 용도에 쓰인다.

PArray를 만드는 방법은 다음과 같다.

- 방출기로 리용하려는(혹은 폭파시키려는) 모형을 만든다.
- Particle Systems에서 PArray를 찰칵한다.
- 작업창문에서 마우스를 끌기하여 PArray를 만든다.
- Basic Parameters목록에서 Pick Object단추를 찰칵한다.
- 만든 모형을 선택한다.

PArray에는 크게 8개의 파라메터설정목록이 있는데 그가운데 몇가지를 보면 다음과 같다.

① Basic Parameters(기본파라메터)

② Particle Generation(립자발생)

Particle Quantity(립자분량)항목에서는 립자가 생성되는 량을 결정한다. Use Rate(사용정도)에서는 매 프레임마다 생성할 립자수를 설정한다. Use Total(사용총수)에서는 총체적으로 발생되는 립자수를 설정한다.

Particle Motion(립자운동)항목에서는 립자들의 운동을 조절한다. Speed(속도)에서 립자들의 속도를 설정하며 Variation(변화)에서 립자들의 속도변화를 설정한다.
Diver gence(분리)에서는 립자들이 날아가는 폭을 설정해준다.

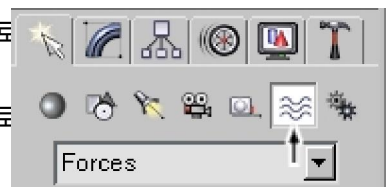
Particle Timing(립자시간지정)항목에서는 립자들의 생성과 소멸을 조절한다. Emit Start(방출시작)에서 방출이 시작되는 첫 순간을 설정한다. Emit Stop(방출 중지)에서는 방출이 끝나는 마지막순간을 설정한다. Display Until(보이기끝)에서는 모든 립자들이 사라지 ... 에서는 립자들 ... 빨리

Particle Types항목에서는 생성되는 립자들의 종류를 선택한다.

Standard(표준)를 선택하면 일반적인 립자들을 만든다. Meta Particles(변성립자)를 선택하면 흐르는 물과 같은 립자들을 만든다. Object Fragments를 선택하면 모형밖에 파편을 만든다. Instanced(실례)를 선택하면 가정된 모형을 립자로 만든다.

Standard Particles(표준립자)항목에서는 시각화할 때 나타나는 립자들의 모양을 설정한다. Triangle(3각형)을 선택하면 3각형으로 만든다. Cube(6면체)를 선택하면 6면체로 만든다. Special(전문적인)을 선택하면 3개의 4각형을 겹쳐서 만든다. Facing(맞은면)을 선택하면 4각형으로 만들되 시각화될 때 항상 정면으로 택

하면 원형으로 만든다. Tetra(3각뿔)를 선택하면 3각뿔로 하

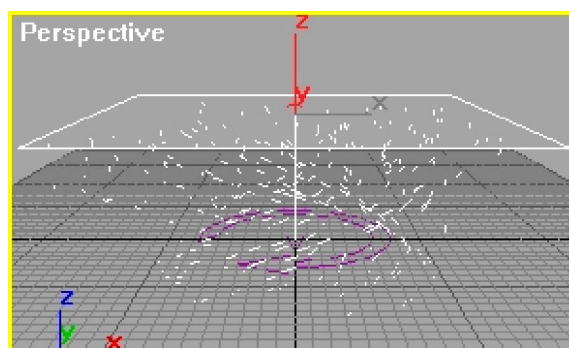
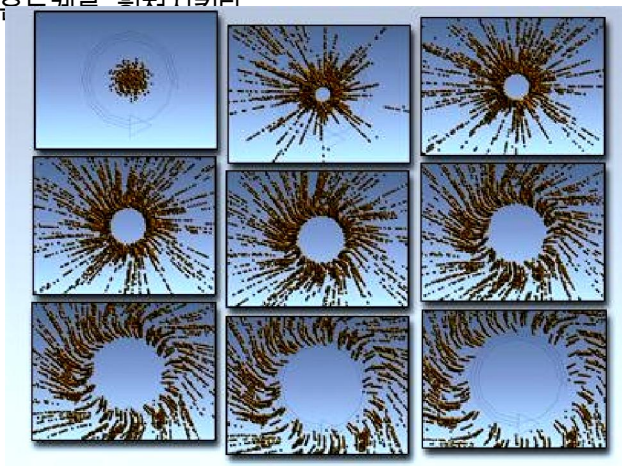
면 별모양으로 만든다. Sphere를 선택하면 구모양으로 만든다.

6) **Pcloud** - 구름이나 어항속의 공기방울 등을 표현할 때 주로 사용한다.

창작판의

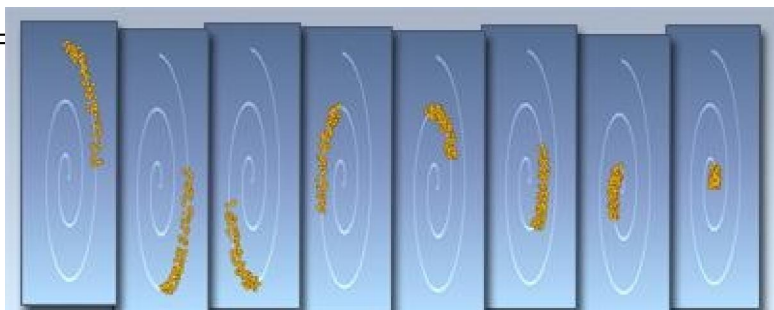
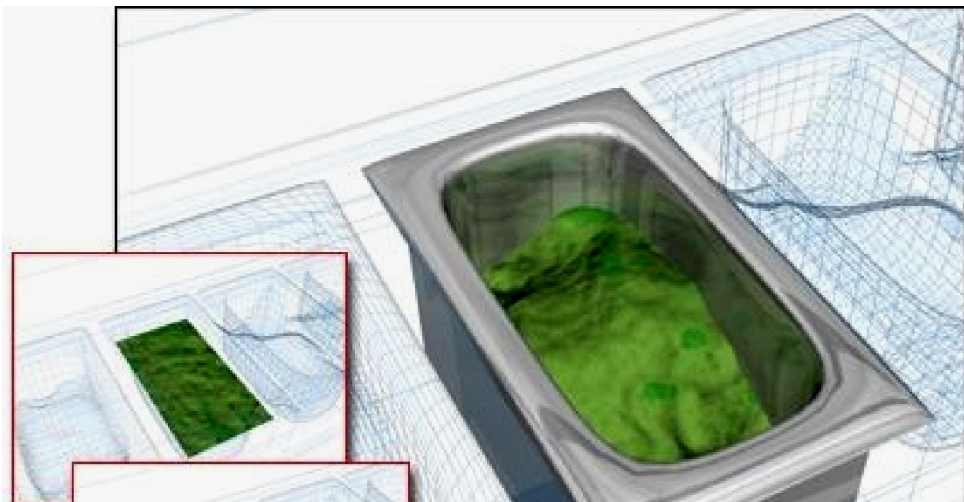
Space W

그러면 Push모형이 생긴다. 다음 도구띠에서 (Bind to Space Warp: 꼬임공간에 묶기)를 선택한 다음 마우스로 Push모형을 끌어서 적용하려는 립자에 가져간다. 그러면 그 립자에 Push가 적용된다.

② **Motor**(원동기)

립자체계나 운동계를 회전시킨다.

⑥ Path Follow(경로추적)

Path F

1

⑨ Displace(옮기기)

콤퓨터

(중학교 제 5 학년용)

제 2 판

집필 리은경, 라순희, 리영걸, 김철호

심사 심의위원회 편집 김복남

장정 리영걸 교정 오혜란

낸곳 교육도서출판사

인쇄소 평양고등교육도서인쇄공장

1판발행 주체98(2009)년 11월 9일

2판인쇄 주체100(2011)년 10월 7일 2판발행 주체100(2011)년 10월 17일

교-11-보-242 값 10 원